Fafoom -Flexible algorithm for optimization of molecules

26.03.2015

Fafoom is a Python module for optimization of organic molecules primarily intended to work with FHI-aims (Fritz Haber Institute ab initio molecular simulations package). Fafoom can be utilized for, e.g., performing a genetic algorithm (GA) search for molecules. The genetic operations (crossover and mutation) explore the fitness function (energy) by changing the torsions of the molecule.

1 Requirements

For the python module:

- Python 2.7
- Numpy
- RDKit (used version: Release_2014_09_2)

For the first-principles methods:

- (recommended) FHI-aims (Fritz Haber Institute ab initio molecular simulations package)
- (alternative) NWChem (NWChem: Open Source High-Performance Computational Chemistry)

2 Installation

1. Clone the fafoom repository

   ```
   git clone https://github.com/adrianasupady/fafoom
   ```

2. Add the fafoom directory to your PYTHONPATH
3. Import the module in python

   ```
   import fafoom
   ```

3 General comments to the examples

The provided example parameter file can be used to run a GA search for alanine dipeptide (Figure 1). Note, that the SMART pattern definitions (torsions, cistrans and custom) are adjusted for this systems so that the peptide bond is treated in a cis/trans mode.
If you want to modify settings or perform a search for another molecule get familiar with section **Keywords**. All the structures generated in course of the algorithm will be saved in the blacklist folder. Moreover, backup text files for the restart are created after each completed iteration. The details about the run are written to `output.txt`.

4 Example: Genetic algorithm with first-principles via FHI-aims

The first example is intended for FHI-aims users. The fafoom package will perform the genetic algorithm and call FHI-aims for local optimization of the 3D structures. Before running the algorithm, the user needs to create a directory containing the `control.in` file that will be used for the FHI-aims calculations. The parameter file needs to contain the command needed to call FHI-aims. The GA can be started with:

```
python ga_simple.py parameter_file
```

One new directory is created for each FHI-aims calculation.

5 Example: Genetic algorithm with force-fields

You can invoke a GA utilizing force-fields with:

```
python ga_ff_simple.py parameter_file
```

Force fields are available from RDKit: UFF and MMFF94 can be chosen. You can also set the convergence settings for the geometry minimization with the force field.

6 Example: Genetic algorithm with first-principles via NWChem

You can invoke a GA utilizing first-principles via NWChem with:

```
python ga_nwchem_simple.py parameter_file
```

If you want to modify the geometry optimization settings, adjust the pynwchem.py file.

```
###### Molecule ######

smile CC(=O)N[C@H](C(=O)NC)C
custom True
smart_torsion [C,N,O]~[!$(*#*)&!D1]-&!@[!$(*#*)&!D1]~[C,N,O]
smart_cistrans C~[$(C=O)]-[$(NC)]~[C]
smart_custom C~[$(C=O)]-[$(NC)]~[C]
rmsd_type cartesian
distance_cutoff_1 1.55
distance_cutoff_2 2.15
rmsd_cutoff_uniq 0.1
chiral True

##### GA settings ####

energy_var 0.001
selection roulette_wheel
fitness_sum_limit 1.2
popsize 5
prob_for_crossing 0.95
prob_for_mut_cistrans 0.4
prob_for_mut_torsions 0.8
max_mutations_cistrans 1
max_mutations_torsions 2
cistrans1 0
cistrans2 180

#####Run settings#####

max_iter 20
iter_limit_conv 10
energy_diff_conv 0.001
energy_wanted -10000000.00
black_dir blacklist

### for FHI-aims:
sourcedir adds
aims_call mpirun -n 4 aims.072713.scalapack.mpi.x
### for force fields:
force_field mmff94
steps 1000
force_tol 1e-04
energy_tol 1e-06
### for NWChem:
functional xpbe96
basis_set STO-6G
nwchem_call mpirun -n 4 nwchem
```

Figure 1: **parameters.txt**

7 Keywords

This section provides the description of parameters and settings.

Molecule settings

- **smile**

  Simplified one-line notation (SMILES) of the compound you want to perform the search for.

- **custom**

  If set to True, you can customize your torsion selection.

- **smart_torsion**

  Pattern for matching torsions.

- **smart_cistrans**

  Pattern for matching *cis/trans* bonds.

- **smart_custom**

  Used only if **custom** is set True. The pattern defined here will be used to match torsions you want to ignore.

- **distance_cutoff_1**

  Parameter for the geometry check. If two non-bonded atoms are closer to each other than **distance_cutoff_1** (Å) the structure will be rejected.

- **distance_cutoff_2**

  Parameter for the geometry check. If two bonded atoms are further from each other than **distance_cutoff_2** (Å) the structure will be rejected.

- **rmsd_type**

  You can decide between *cartesian* and *torsional* RMSD to be used for distinguishing between similar and different structures. If *cartesian* is chosen, the GetBestRMS RDKit routine will be used for calculating the RMSD between two structures. The *torsional* RMSD might be quicker than the *cartesian* RMSD, but is not symmetry corrected. However, you can adapt the get_vec function (in the utilities module) for your needs.

- **rmsd_cutoff_uniq**

  This parameter is used for blacklisting. The unit depends on the type of the RMSD selected in **rmsd_type** (Å for *cartesian* and rad for *torsional*). A new structure is considered to be unique if it has an RMSD to all already existing structures higher than the **rmsd_cutoff_uniq**. If you set the threshold to 0.0 all structures will be treated as unique. However, they will still be stored in the blacklist directory.

- **chiral**

  If set to False, not only the structure but also its mirror image will be used for comparisons.

GA settings

- **popsize**

  Size of the initial pool of structures.

- **energy_var**

  If the difference between the highest and lowest energy in the population is lower than the **energy_var**, all the individuals will be assigned the same fitness of 1.0

- **selection**

  Options for the selection mechanisms of the individuals. Another options are random and roulette_wheel_reverse .

- **fitness_sum_limit**

  If the sum of the fitness values for all individuals is lower than this threshold the selection will be conducted independently from the chosen mechanism. The best and a random individual will be selected.

- **prob_for_crossing**

  Probability for the crossing over.

- **prob_for_mut_cistrans**

  Probability for a mutation in *cis/trans* bonds.

- **prob_for_mut_rot**

  Probability for a mutation for the torsions

- **max_mutations_cistrans**

  Maximal number of mutations for *cis/trans* bonds. A random number between (1, **max_mutations_cistrans**) of mutations will be performed.

- **max_mutations_torsions**

  Maximal number of mutations for torsions. A random number between (1, **max_mutations_torsions**) of mutation will be performed.

- **cistrans1**

  First option for *cis/trans* bond value during population initialization.

- **cistrans2**

  Second option for *cis/trans* bond value during population initialization. If the value is equal to the value of **cistrans1** only cis (0) or trans (180) conformations will be evaluated.

Run settings

- **max_iter**

  Number of iterations that will be performed after the initialization is finished.

- **iter_limit_conv**

  Minimal number of iterations to be performed before any convergence criteria are checked.

- **energy_diff_conv**

  Parameter for checking the convergence. If the lowest energy hasn't change by more than **energy_diff_conv** (eV) after **iter_limit_conv** iterations, the GA-run is considered to be converged. Attention: convergence doesn't necessarily mean that the global minimum was found.

- **energy_wanted**

  If the energy of the global minimum is known it can also be used for checking if the convergence is achieved. It should be set low enough if the energy is not known.

- **black_dir**

  Name of a directory that will be created to keep the blacklisted structures.

- **(FHI-aims) sourcedir**

  Name of your directory with control.in file.

- **(FHI-aims) aims_call**

  String for execution of FHI-aims.

- **(FF from RDKit) force_field**

  Name of the force field to be used.

- **(FF from RDKit) steps**

  Number of steps for the minimization.

- **(FF from RDKit) force_tol**

  Force tolerance.

- **(FF from RDKit) energy_tol**

  Energy tolerance.

- **(NWChem) functional**

  Functional to be used.

- **(NWChem) basis_set**

  Basis set to be used.

- **(NWChem) nwchem_call**

  String for execution of NWChem.

8 General advice

- take your time to construct and test a reasonable control.in file
- be careful when adjusting the **distance_cutoff** parameters
- adjust the **smart_torsion** and **smart_cistrans** to your needs; check if the recognized torsions are fine for you

9 Ongoing

- symmetry correction in the torsional RMSD
- optimization of shared blacklisting