J Mol Model (2001) 7:374–377
DOI 10.1007/s00894-001-0051-z

ORIGINAL PAPER

Thomas Strassner · Markus Busold · Helmuth Radrich

FFGenerAtor 2.0 – an automated tool for the generation of MM3 force field parameters

Received: 17 July 2001 / Accepted: 6 September 2001 / Published online: 20 October 2001

Abstract *FFGenerAtor 2.0* is a tool to customize the MM3 force field. It consists of two main programs, one that determines the missing parameters in the chosen structures and one that optimizes the parameter set using a genetic algorithm.

The C++ program was developed on a LINUX system; all necessary software is available free of charge. The best parameter set is determined without changing the original MM3 parameters based on the chosen structures. Several different switches allow the properties and composition of the genetic algorithm to be changed.

Keywords MM3 · Genetic algorithm · Force field · Parameterization

Introduction

Force field calculations are mainly used for the calculation of large structures of organic and biological molecules and to study a high number of conformations. The advantage of force fields lies in their speed: they are several orders of magnitude faster than semiempirical or ab initio calculations and can therefore handle a much larger number of atoms. They can be applied routinely and for most researchers it is not necessary to customize force fields for their own needs because the existing force fields [1] have been parameterized for the most common elements.

However, in some areas such as organometallic chemistry, metalloproteins containing transition metals or very specific structures, parameters are still missing.

Several approaches towards a solution of that problem have been published before. The UFF force field [2] includes all elements and allows calculations on any type of compound. Others have pursued approaches to parameterizing a specific force field for a certain type of complex [3, 4, 5, 6] or even for transition states. [7] The data for these parameterizations were mostly derived from accurate high-level DFT or ab initio calculations.

However, this process is time consuming and impractical if you are missing a specific parameter for your problem. Therefore, we wanted to develop a highly automated process that generates good-quality parameters. Due to the multidimensional parameter hypersurface and the inherent dependency of the parameters, which have to be optimized, genetic algorithms seem to be the most promising approach.

FFGenerAtor 2.0 is a tool to create specific MM3 parameters from available data (computational or X-ray) based on a genetic algorithm that only needs the user input of the structures to be included in the parameterization.

Genetic algorithms (GA) are a quite common optimization technique used in various fields of science. [8, 9, 10, 11] In the last few years they have been applied to many chemical problems, especially in the field of drug design. [12]

From the available force fields we chose MM3, [13, 14, 15] an established and reliable method, which is widely used for force field calculations, but with some minor modifications our tool can also be used for the generation of other force field parameters.

All programs used are available free of charge; the tool was written in C++ and developed on a Linux platform running SuSE 6.3. FFGenerAtor can be obtained at the following Url: http://www.compchem.de/ffgenerator.

Computational details

FFGenerAtor 2.0 consists of two main programs, which are both highly automated: one checks for missing parameters and provides the data for the other program, which optimizes them without changing the original parameter set.

T. Strassner (✉) · M. Busold · H. Radrich
Anorganisch-Chemisches Institut,
Technische Universität München, Lichtenbergstr. 4,
85747 Garching b. München, Germany
e-mail: Thomas.Strassner@ch.tum.de
Tel.: +49-89-28913174, Fax: +49-89-28913473

Force field calculations

The MM3 force field is implemented in many packages. We chose the *TINKER 3.7* program package, [16] because of its very large flexibility concerning the choice of force field or method. It also provides the capability of molecular dynamics calculations and can be used together with Gamess for QM/MM calculations. [17]

To treat structures with coordination numbers higher than 4, e.g. metal–organic complexes, it is necessary to modify the parameter *maxval* and recompile the program. The necessary modifications are described in the *TINKER* documentation.

Genetic algorithm

GALib 2.4.5 [18] (http://lancet.mit.edu/ga), a C++ library of genetic algorithm objects has been used to implement the basic GA functionality. The library includes several different tools for optimization using various representations and genetic operators. *FFGenerAtor 2.0* uses a real number genome to represent the variables of the force field. Elitism is mandatory as well as a linear scaling scheme.

Two types of genetic algorithms (SimpleGA and SteadyStateGA) together with two different selection schemes (Roulette Wheel Selection and Tournament Selection) can be chosen. The crossover and mutation rates are also variable.

In a study [19] on the parameterization of rhenium compounds using a previous version of FFGenerAtor we found that the best results can be obtained for the combination of a SteadyState GA with a Tournament Selector using a crossover rate of 0.9 and a mutation rate of 0.02.

Structural data for the parameterization

Any data can be used for the parameterization, whether they are quantum mechanically calculated structures or CSD data. But a force field can only be as good as the underlying data set for the parameterization, so the choice is critical for the quality of the resulting parameters. The choice of structures should intend to have as many different parameters as possible without biasing the GA.

Parameter numbering

In the case that a new parameter is created, the user has to choose a new parameter number that may not have been used already. The original MM3 force field parameter list contains approximately 150 parameters and some programs are not able to handle parameter numbers larger than 500. Therefore, for the sake of compatibility the new parameters should be in the range between 200 and 500.

Input creation and visualization

To create the input files for *TINKER* [16] and to visualize the results, the graphical interface *MOLDEN* [20] is recommended. *MOLDEN* is able to read and write several formats, of which we suggest the mol2 format for the exchange of the coordinates between the CSD and *MOLDEN*.

Fitness function

Structures before and after the minimization are compared by *TINKER*'s *superpose*, which calculates the rms deviation (in mass- or unit-weighted coordinates) between two structures. The resulting rmsd between the input data and the force-field optimized structure is then used to generate a fitness value for each parameter set. This approach has been applied successfully before, [21, 22] but it must be noted that this particular kind of fitness function has advantages and disadvantages. A very high level of automation requiring only minimal user input can be achieved most easily using the geometrical deviation as a fitness function. One disadvantage is that spectroscopic data cannot be included for the parameterization, but the main goal of this program was a fast and highly automated approach for the generation of force fields.

The quality of the force field created can also be determined by an external validation test set of structures that are not part of the parameterization data. [6]

Genetic algorithm optimizer

The force field parameters are represented by chromosomes of real numbers. The chromosome of each individual in a population is built from genes that correspond to the parameters for bonds, angles, torsions and out-of-plane bending. The parameters for bond lengths and angles can be configured to optimize in a user-defined interval; the recommended default criteria are ±0.5 Å for bonds and ±30° for angles around the average values for these parameters in the sample structures. The default constraints for force constants are: bond stretching 0.0–250.0 mdyn/Å, angle bending 0.0–30.0 mdyn/rad^2, torsions –200.0 to +200.0 kcal/mol. These quite high and uncommon boundaries are chosen to ensure that the GA can optimize all values freely, but the user can set his own criteria.

Hardware requirements

All calculations can be performed on standard PCs running Linux (the program was run successfully on SuSE Linux 6.4 and RedHat 7.0). The time for the optimization depends on several criteria, like the CPU speed, the number of structures in the parameterization, the popula-

Table 1 Example of FFGA.ini – the main configuration file of *FFGenerAtor 2.0*

```
[GA]
gaNnGenerations     150
gaNpopulationSize   100
gaNminimaxi         0
gaNscoreFrequency   1
gaNflushFrequency   1
gaNrecordDiversity  true
gaNscoreFilename    myproblem_out.dat
gaNpCrossover       0.9
gaNpMutation        0.02
Path                /path/to/my/workdir/temp2/
FilesINI            p2files.ini
GaType              0
CBGAdiffFile        myproblem_diffcbga.out
BAKdiffFile         myproblem_diffbak.out
KeyFileMAXITER      2500

 [Undefined Bond Stretching Parameters]
0 50 0.8 3.0
     +5-Ring
     0 20 1.0 2.80
          >114 113
          0 50 0.85 1.20

[Undefined Angle Bending Parameters]
0 50 30 180

[Undefined Torsional Parameters]
-50 50    -50 50    -50 50

[Undefined Out-of-Plane Bending Parameters]
-200 200
```

tion size, the number of generations and the size of the interval of bond lengths and angles. As an example: a data set of ten organometallic compounds with a population size of 100 and 150 generations (optimization steps) takes about 1–2 days on a 650 MHz Athlon processor.

Configuration of *FFGenerAtor 2.0*

FFGenerAtor 2.0 is configured through its configuration file FFGA.ini (Table 1). The program has a large number of configuration options that will be described and explained in the following text. “gaNnGenerations” sets the number of generations of the GA, which can be seen as the number of optimization steps, typically between 100 and 200. “gaNpopulationSize” regulates the population size, which means the number of parameter sets in each generation, generally values of 75–200. “gaNminimaxi” decides if the fitness criterion is either minimized or maximized, which obviously has to be set to minimization. “gaNscoreFrequency”, “gaNflushFrequency”, “gaNrecordDiversity” and “gaNscoreFilename” adjust how often and which statistical data are recorded and to which filename they are written. “gaNpCrossover” and “gaNpMutation” set the crossover and mutation rates. Recommended crossover rates are between 0.5 and 0.9, mutation rates between 0.01 and 0.20. “Path” tells the program where the optimization will be executed. This is normally a temporary directory or preferably a path to a ramdisk, which improves the performance significantly as all operations are then carried out in memory. The “FilesINI” tells the program the name of a file, which has to be created by the user. It contains the names of the files that have been chosen as the “data base” for the parameterization. “GaType” allows the user to choose the GA, currently four options are available (SimpleGA/Tournamentselector=0, SimpleGA/Rouletteselector=1, SteadyStateGA/Tournamentselector=2, SteadyStateGA/Rouletteselector=3). “CBGAdiffFile” and “BAKdiffFile” specify the filenames where the changes between the different generations are recorded with respect to the best parameter-sets. “KeyFileMAXITER” sets a maximum number for the MM3 optimization steps for *TINKER*’s optimize. To increase performance, typical values are between 1,500 and 2,000 because bad parameter sets would waste time and optimize too long. The last section is completely dedicated to the setting of boundaries for the force field parameters that are optimized. An example of the possibilities is given in the section ‘‘[Undefined Bond Stretching Parameters]’’. The first row tells the program that the force constant should be optimized between 0 and 50 mdyn/Å and that the equilibrium bond length between 0.8 and 3.0 Å. These values are then applied to all bonds unless there are more specific instructions like the ones that follow now. ‘‘+5-Ring” states that the next values are applied to all bonds in a 5-Ring. Here all 5-Ring bonds are allowed to optimize in an interval of 0–20 mdyn/Å for the force constants and 1.0–2.8 Å for the equilibrium bond lengths. ‘‘>114 113” is even more specific as this rule applies then to all 5-Ring bonds between atom types 114 and 113. This procedure can also be applied to Angles, Torsions and Out-Of-Plane Bending parameters, allowing the user to set very specific values for any parameter.

Conclusions

FFG*ener***A***tor 2.0* is an automated program for the parameterization of MM3 **f**orce **f**ields on the basis of a **g**enetic **a**lgorithm. From quantum chemical and X-ray structures missing parameters are identified automatically. The optimization of these parameters is then carried out with a large range of user-defined parameters that tweak the GA as well as optimization boundaries for the parameters.

Acknowledgements We are indebted to Prof. W.A. Herrmann for his generous and continuous support of our work. Grants of the Dr. Karl-Wamsler-Stiftung and the Leonhard-Lorenz-Stiftung are gratefully acknowledged.

References

1. Young DC (2001) Computational chemistry: a practical guide for applying techniques to real world problems. Wiley, New York
2. Rappe AK, Colwell KS, Casewit CJ (1993) Inorg Chem 32:3438
3. Cundari TR (2001) Computational organometallic chemistry. Marcel Dekker, New York
4. Norrby PO, Brandt P (2001) Coord Chem Rev 212:79
5. Hagelin H, Akermark B, Norrby PO (1999) Organometallics 18:2884

6. Hagelin H, Svensson M, Akermark B, Norrby PO (1999) Organometallics 18:4574
7. Eksterowicz JE, Houk KN (1993) Chem Rev (Washington, D.C.) 93:2439
8. Judson R (1997) Rev Comput Chem 10:1
9. Mitchell M (1996) An introduction to genetic algorithms. MIT Press, Cambridge, Mass.
10. Goldberg DE (1989) Genetic algorithms in search, research and machine learning. Addison-Wesley, New York
11. Holland J (1975) Adaption in natural and artificial systems. MIT Press, Mich.
12. Douguet D, Thoreau E, Grassy G (2000) J Comput-Aided Mol Des 14:449
13. Lii JH, Allinger NL (1989) J Am Chem Soc 111:8566
14. Lii JH, Allinger NL (1989) J Am Chem Soc 111:8576
15. Allinger NL, Yuh YH, Lii JH (1989) J Am Chem Soc 111:8551
16. Ponder JW (1999) TINKER 3.7
17. WWW: http://www.arl.hpc.mil/PET/cta/ccm/arl-tips/gamess/simomm.html
18. Wall M (1999) GALib 2.4.5
19. Strassner T, Busold M, Herrmann WA (2001) J Comput Chem accepted
20. Schaftenaar G, Noordik JH (2000) J Comput-Aided Mol Des 14:123–134
21. Hunger J, Beyreuther S, Huttner G, Allinger K, Radelof U, Zsolnai L (1998) Eur J Inorg Chem 6:693
22. Hunger J, Huttner G (1999) J Comput Chem 20:455