

***ConfSearch*: A conformational search program**

Manual

Rubén Meana-Pañeda and Donald G. Truhlar

Department of Chemistry, Chemical Theory Center, and Supercomputing Institute

University of Minnesota, Minneapolis, MN 55455

Program version: 2015

Program version date: July 16, 2015

Manual updated date: July 17, 2015

Copyright 2015

Abstract: *ConfSearch* is a computer program for performing conformational searches by rotating bonds and using the rotated structures as starting points for optimization.

Table of contents

1. Introduction	2
2. User agreement	3
3. Citation	3
4. Distribution and installation	4
5. Workflow of <i>ConfSearch</i> program	6
6. Input file	7
7. Output files	9
8. Test runs for <i>ConfSearch</i>	9
9. Computers, operating systems, and compilers on which the code has been tested	
10. Acknowledgments	9
11. Revision history	10

1. Introduction

The *ConfSearch* program is a FORTRAN90 computer program for conformational searching. The program generates a set of conformational structures by rotating user-specified bonds in an input structure and optimizing the structures by using *MOPAC*,¹ *MOPACmn*,² *Gaussian03*, or *Gaussian09*³ software package. The program can locate both, minima and transition structures by using the geometry optimization methods that are available in the electronic structure software package that is used.

1 Stewart, J. J. P. *MOPAC2012*; Stewart Computational Chemistry; Colorado Springs, CO.
<http://openmopac.net/MOPAC2012.html>

2 Stewart, J. J. P. et al., *MOPAC 5.022mn*; University of Minnesota, Minneapolis, MN.
<http://comp.chem.umn.edu/mopac>

3 Frisch, M. J. *et al.* *Gaussian 03/09*; Gaussian, Inc.: Wallingford, CT.
http://www.gaussian.com/g_prod/g09.htm

2. User agreement

ConfSearch is a licensed program, and the use of this program implies acceptance of the terms of the license, which are repeated here for convenience:

A. No user or site will redistribute the source code or executable code to a third party in original or modified form without written permission of the principal investigator (Donald G. Truhlar). A license does not entitle the licensee to relicense the code or distribute it in original or modified form to parties not covered by the license. The licensee has no ownership rights in the *ConfSearch* software or in any copyrights for the *ConfSearch* software or documentation through this license. A user license covers the work of a single research group and the code may be shared and disseminated within a group without requiring permission. Site licenses are available with a different license (see *ConfSearch* home page).

B. Publications resulting from using this package will cite the corresponding program. The required reference is given in the documentation.

C. No guarantee is made that this program is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within the program.

D. No consulting or maintenance services are guaranteed or implied.

3. Citation

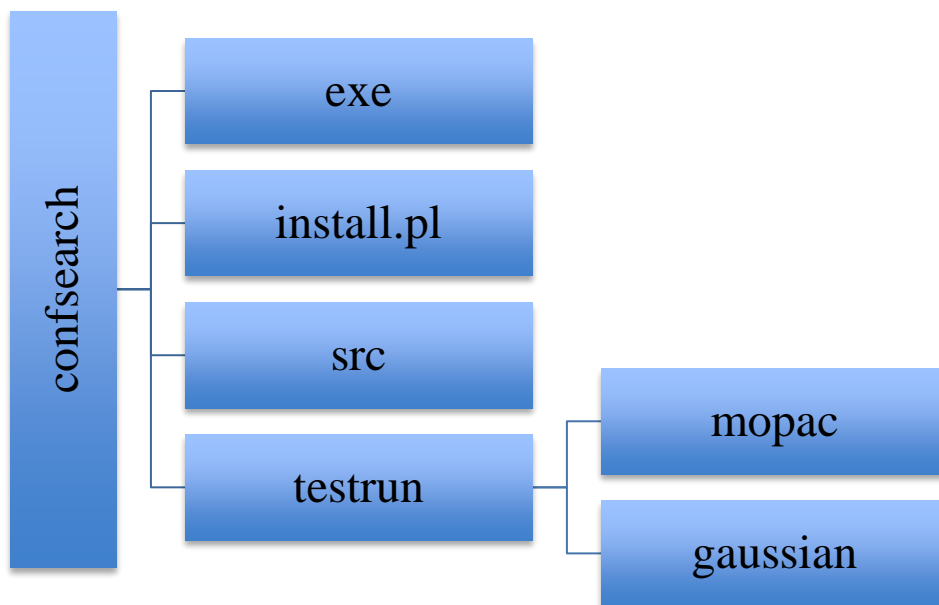
Publications including work performed with *ConfSearch* should cite the software, for example, as follows:

R. Meana-Pañeda and D. G. Truhlar *ConfSearch* – version 2015, University of Minnesota, Minneapolis, 2015. <http://comp.chem.umn.edu/confsearch>

4. Distribution and installation

4.1. Distribution

The distributed *ConfSearch* package, *confsearch.tar.gz*, contains the following directories and files:



exe: This is an empty directory. After successful installation, this directory should contain the *ConfSearch* executable file, *confsearch.exe*.

install.pl: This is a Perl script that can be used for installation of the *ConfSearch* program.

src: This directory contains the source code of the program.

testrun: This directory contains two subdirectories, *mopac* and *gaussian*, depending of the electronic structure package that is used. Each subdirectory has one example test file for the 2-heptyl radical.

4.2. Installation

The installation procedure consists of the following steps:

- 1) Copy the *confsearch.tar.gz* package to the installation directory and then unzip and untar the package.

```
tar -xzf confsearch.tar.gz <Return>
```

- 2) Go to the *confsearch/* directory and run the installation script

```
./install.pl <Return>
```

The *install.pl* script will search for available Fortran compiler and compile the *ConfSearch* code in the *confsearch/src/* directory.

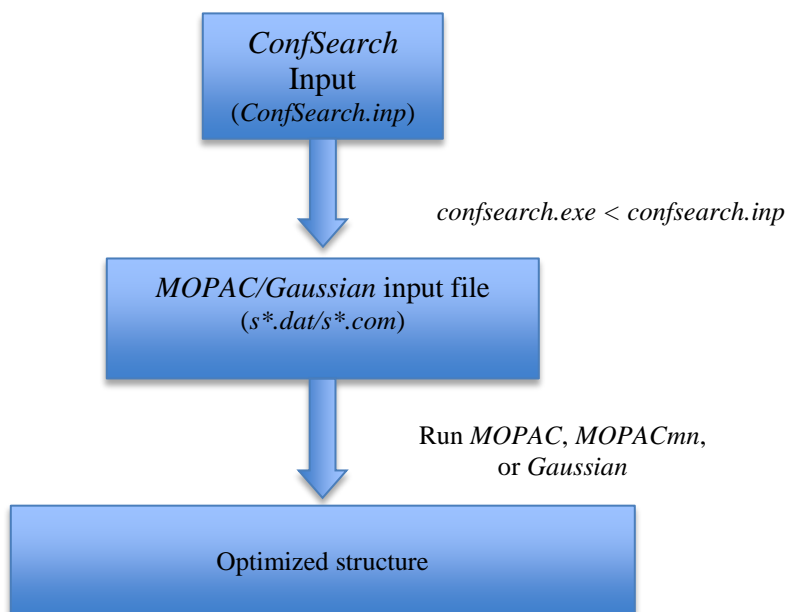
- 2) If the install script runs successfully, the *confsearch/exe/* directory will contain the executable: *confsearch.exe*
 - 3) The program should be installed on a system on which *Gaussian09*, *Gaussian03*, *MOPAC*, and/or *MOPACmn* runs correctly because *ConfSearch* calls *Gaussian09*, *Gaussian03*, *MOPAC*, or *MOPACmn* to carry out the geometry optimization.
 - 4) Add the *confsearch/exe/* directory to your Linux/Unix PATH variable.
- Alternatively, one can copy the executable to a directory that is already in the Linux/Unix PATH variable.

4. 3. Description of Executable

Executable	Description	Usage
<i>confsearch.exe</i>	Code to generate a set of conformational structures by rotating user specified bonds in an input structure and optimizes them by using an electronic structure software package.	<i>confsearch.exe</i> < input

5. Workflow of *ConfSearch* program

This section describes the steps that *ConfSearch* takes to do the conformational search in combination with the *MOPAC*, *MOPACmn*, or *Gaussian* software package. The following flowchart shows the process:



In first step, the program generates the *MOPAC*, *MOPACmn*, or *Gaussian* input file with the initial structure generated by rotating one or more bonds. In a second step, *ConfSearch* optimizes the structure by running the electronic structure package (*MOPAC*, *MOPACmn*, or *Gaussian09*, or *Gaussian03*). These two steps are repeated until all the structures are optimized.

6. Input file

The executable *confSearch.exe* generates a list of conformational structures by rotating a set of user-specified bonds of an input structure over a specified grid. The program writes each generated structure to a file in *MOPAC/MOPACmn/Gaussian* input format and optimizes the geometry of the structure.

Example for MOPAC or MOPACmn:

```

19 3 ! number of atoms and number of rotating bonds
C   -10.440384 -1.363155 -2.523127 !Cartesian coordinates
H   -10.785179 -0.329456 -2.469970
H   -10.859509 -1.809586 -3.424204
H   -6.394624 -0.330209 -2.139857
C   -8.917768 -1.426103 -2.518649
H   -8.590394 -2.466364 -2.596066
H   -8.525182 -0.915036 -3.401850
C   -8.310574 -0.801852 -1.268354
H   -8.640182  0.237649 -1.182536
H   -8.692678 -1.318982 -0.382639
C   -6.785578 -0.845494 -1.253867
H   -6.459804 -1.891949 -1.368236
C   -6.182441 -0.253469 -0.029715
H  -10.853238 -1.892735 -1.663274
H   -6.763400 -0.260765  0.883956
C   -4.722319  0.016971  0.039075
H   -4.137495 -0.912453  0.059618
H   -4.453027  0.588421  0.925756
H   -4.383850  0.571246 -0.840698
#torsion 1 definition
5 8 !define rotation axis as the bond between atom 5 and atom 8
7 !number of atoms in one fragment
1 2 3 14 5 6 7 ! atom numbers of this fragment
12 ! Grid of points in the interval [0-360] degrees
#torsion 2 definition
8 11 !define rotation axis as the bond between atom 8 and atom 11
10 !number of atoms in one fragment
1 2 3 14 5 6 7 8 9 10 ! atom numbers of this fragment
12 ! Grid of points in the interval [0-360] degrees
#torsion 3 definition
11 13 !define rotation axis as the bond between atom 11 and atom 13
13 !number of atoms in one fragment
1 2 3 14 5 6 7 8 9 10 11 4 12 ! atom numbers of this fragment
12 ! Grid of points in the interval [0-360] degrees
~/soft/MOPAC/MOPAC2012.exe ! location of the MOPAC code executable
PM7 PRECISE ! keywords of the input file

```

Example for Gaussian03 or Gaussian09:

```

19 3 ! number of atoms and number of rotating bonds
C -10.440384 -1.363155 -2.523127 !Cartesian coordinates
H -10.785179 -0.329456 -2.469970
H -10.859509 -1.809586 -3.424204
H -6.394624 -0.330209 -2.139857
C -8.917768 -1.426103 -2.518649
H -8.590394 -2.466364 -2.596066
H -8.525182 -0.915036 -3.401850
C -8.310574 -0.801852 -1.268354
H -8.640182 0.237649 -1.182536
H -8.692678 -1.318982 -0.382639
C -6.785578 -0.845494 -1.253867
H -6.459804 -1.891949 -1.368236
C -6.182441 -0.253469 -0.029715
H -10.853238 -1.892735 -1.663274
H -6.763400 -0.260765 0.883956
C -4.722319 0.016971 0.039075
H -4.137495 -0.912453 0.059618
H -4.453027 0.588421 0.925756
H -4.383850 0.571246 -0.840698
#torsion 1 definition
5 8 !define rotation axis as the bond between atom 5 and atom 8
7 !number of atoms in one fragment
1 2 3 14 5 6 7 ! atom numbers of this fragment
12 ! Grid of points in the interval [0-360] degrees
#torsion 2 definition
8 11 !define rotation axis as the bond between atom 8 and atom 11
10 !number of atoms in one fragment
1 2 3 14 5 6 7 8 9 10 ! atom numbers of this fragment
12 ! Grid of points in the interval [0-360] degrees
#torsion 3 definition
11 13 !define rotation axis as the bond between atom 11 and atom 13
13 !number of atoms in one fragment
1 2 3 14 5 6 7 8 9 10 11 4 12 ! atom numbers of this fragment
12 ! Grid of points in the interval [0-360] degrees
g09 ! Gaussian09 or Gaussian03
%nproc=8 ! number of processors for Gaussian job
%mem=1000MB ! memory
# PM6 opt ! route section
0 2 !charge and multiplicity
Extbasis ! !line at the end of the Gaussian input file

```

It should be noted that this utility code removes all the generated structures that show an interatomic distance smaller than 0.5 Å.

7. Output files

In the current directory the program generates the *MOPAC*, *MOPACmn*, or *Gaussian* input file and the geometry optimization output file of every structure. Furthermore, the code creates a file called *geom.xyz* that contains the Cartesian coordinates of all the initial structures, before they are optimized.

8. Test runs for *ConfSearch*

Two subdirectories, called *mopac* and *gaussian*, are included in the *confsearch/testrun* directory. They correspond to the conformational search tests by using *MOPAC* or *MOPACmn* or by *Gaussian03* or *Gaussian09* respectively. Each subdirectory contains one test input file, *2heptyl.inp* for the 2-heptyl radical.

9. Computers, operating systems, and compilers on which the code has been tested

9.1. Version 2015

Computer	OS	Compilers
Itasca ¹	CentOS 6.2	ifort/11.1 and icc/11.1 gfortran/4.4 and gcc 4.4
Mesabi ²	CentOS 6.6	ifort/13.1 gfortran/4.4

1. Itasca is a Linux cluster of HP Proliant BL280c G6 Blade servers.

2. Mesabi is a heterogeneous HP cluster

10. Acknowledgments

This work was supported in part by the U.S. Department of Energy (DOE) Office of Science, Office of Basic Energy Sciences, as part of the Combustion Energy Frontier Research Center under Award Number DE-SC0001198. This work was also supported by the DOE through grant no. DE-FG02-86ER13579.

11. Revision history

Version 2015 (July 17, 2015)

Authors: R. Meana-Pañeda and D. G. Truhlar

This is the first distributed version.