

Generation of Gaussian 09 Input Files for the Computation of ^1H and ^{13}C NMR Chemical Shifts of Structures from a Spartan'14 Conformational Search

Patrick Willoughby (✉ willoughbyp@ripon.edu)

Ripon College

Spencer Reisbick

Ripon College

Method Article

Keywords: NMR Computational Chemistry Organic Chemistry

Posted Date: October 21st, 2019

DOI: <https://doi.org/10.21203/rs.2.1186/v2>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Version of Record: A version of this preprint was published at Protocol Exchange on April 28th, 2014. See the published version at <https://doi.org/10.1038/protex.2014.015>.

Abstract

This protocol describes an approach to preparing a series of Gaussian 09 computational input files for an ensemble of conformers generated in Spartan'14. The resulting input files are necessary for computing optimum geometries, relative conformer energies, and NMR shielding tensors using Gaussian. Using the conformational search feature within Spartan'14, an ensemble of conformational isomers was obtained. To convert the structures into a format that is readable by Gaussian 09, the conformers were first exported to a single ".sdf" file. A Python script was used to (i) read the structural information of each conformer within the ".sdf" file and (ii) write the corresponding atomic coordinates into a series Gaussian 09 input files. This approach decreases the amount of active effort required to compute NMR chemical shifts of a structure that populates an ensemble of conformers.

Introduction

NMR spectroscopy is the most useful tool for determining the structure of an unknown organic molecule. By coupling this approach with other analytical techniques (e.g. mass spectrometry) the structure of an unknown organic molecule can be elucidated. However, molecules of greater complexity continue to be isolated and/or prepared, and their associated analytical data are increasingly convoluted. Consequently, the assigned structures of these newly isolated compounds are sometimes incorrect, which leads to years of misguided effort "chasing molecules that were never there"¹. Modern computational chemistry software packages (e.g., Spartan^{2,3}, Gaussian 09⁴, and Jaguar^{5,6}) have enabled the routine use of density functional theory (DFT) calculations for predicting spectroscopic properties of organic molecules. For example, one of us recently reported a protocol that described an approach using Gaussian 09 to compute NMR data for molecules that adopt conformational isomers⁷. An important, early part of this protocol required the use of the software application, MacroModel⁸ (part of the Schrödinger suite) to carry out a stochastic conformational search using the OPLS molecular mechanics force field. For each structure resulting from this conformational search, free energies and NMR shielding tensors were calculated. Using the free energy data, a Boltzmann factor was determined for each conformer, which was, in turn, converted into the relative mole fraction. The computed NMR data are averaged (using the mole fraction of each conformation), referenced, and scaled to generate a set of Boltzmann-weighted average chemical shifts. Due to the widespread use of Spartan for molecular mechanics calculations, we have prepared an addendum to this protocol that utilizes the structures resulting from a Spartan _Conformer Distribution_ calculation. As discussed in our original protocol,⁷ molecules of increasing complexity are often accompanied by many conformational isomers. We have developed a Python script (e.g., "write-g09-inputs-sdf.py") that generates two Gaussian 09 input files for each structure resulting from the conformational search. For convenience, we have provided an additional script (e.g., "write-g03-inputs-sdf.py") that prepares Gaussian 03 input files. These input files include an "_-opt_freq_" file for determining the optimal geometry and free energy along with an "_-nmr_" file for calculating NMR shielding tensor data. The Python script expedites the DFT computations by greatly simplifying the preparation of the Gaussian input files. More specifically, the script extracts structural information from a ".sdf" file generated in Spartan, and the coordinates of each conformation are written into the Gaussian input files. The ".sdf" file type is routinely used for storing molecular information for multiple structures and can be produced by myriad software applications. The script provided in this protocol will be useful for writing Gaussian input files from ".sdf" files prepared in other chemistry software applications.

Equipment

Software required to carry out Python scripts • Command-line interface application (Terminal in Mac OS X or Linux; or Command Prompt in Windows) • Python, version 2 or 3 (included with Mac OS X and Linux operating systems) • Python script editor (e.g., IDLE (see <http://www.python.org/download/>):<http://www.python.org/download/>)) • Text editing application (e.g., TextEdit in Mac OS X or Notepad in Windows)

Software requirements for calculations • This protocol has been written for use with Spartan'14^{2,3}; however, we have tested earlier versions of Spartan (e.g., Spartan'08) and found that they are also compatible with the following Procedure. • The approach described in the Procedure is amenable to any software application that is capable of performing a conformational search and exporting the family of conformers as a ".sdf" file (e.g., MacroModel⁸ and ChemBio3D⁹).

Hardware requirements for use of Python scripts • Most standard personal computers built after 2008 are capable of executing the Python scripts included in this protocol.

Hardware requirements for conformational search calculation • A CPU with 4 GB of RAM and a dual-core processor is capable of performing the conformational search calculation for generating a family of conformers of the candidate structure. The hardware requirements for carrying out the DFT calculations in Gaussian 09 are described in ref. 7.

Procedure

Create input geometry and carry out a molecular mechanics conformational search in Spartan ● **TIMING** 30 min (Steps 1 – 8)

- Draw cis-3-methylcyclohexanol in the Spartan workspace.** In Spartan select **_File_** → **_New_** to open the **_Model Kit_** toolbar. Change the **_Rings_** drop down menu to **_Cyclohexane_** and click inside the workspace to add a parent cyclohexane molecule. Select the **_Csp3_** button within the **_Model Kit_** toolbar and click one of the yellow open valences on the cyclohexane ring. This operation will attach the methyl group. Select the **_Osp3_** button within the **_Model Kit_** toolbar and click a yellow open valence on the cyclohexane ring that is both two carbons separated from and **_cis_** to the previously added methyl group. This will add an oxygen atom to the cyclohexane ring to give **_cis_-3-methylcyclohexanol**. Ensure that the overall structure is **_cis_-3-methylcyclohexanol** before continuing.
- Quick and Crude Molecular Mechanics Geometry Optimization.** Cleanup the preliminary geometry by clicking the **_Minimize_** button or selecting **_Build_** → **_Minimize_**.
- Perform the conformational search.** Select **_Setup_** → **_Calculations_**. A window will open. Change the **_Calculate:_** drop down menu selection to **_Conformer Distribution_**. Change the two drop down menus to the right so that they display **_Molecular Mechanics_** and **_MMFF_**. Check the box next to **_Maximum:_** and change the conformers examined to "1000". Click **_Submit_** and a **_Save As_** window will appear. Change the computational filename to "cis-3-methylcyclohexanol", change the directory (i.e. folder) to a location that is convenient for storing the associated computational files, and click **_Save_**. Click **_OK_** in the window that appears, which indicates that the conformational search has started.
- After the conformational search has finished, a window confirming that the job has completed will appear. Click **_OK_** in this window. When prompted to open a new document select **_No_**. Select **_File_** → **_Close_** to clear the workspace.
- Select **_File_** → **_Open_** and locate the conformational search output file "cis-3-methylcyclohexanol.spartan."
- Ensure that all expected conformers have been found by the calculation. Select **_Display_** → **_Spreadsheet_** to open a window containing an entry for each structure found during the conformational search. A conformational search of **_cis_-3-methylcyclohexanol** is expected to yield six unique conformers.
- Export the library of conformers as a single ".sdf" file. Select **_File_** → **_Save As..._** to open

a `_Save As_` window. Enter "cis-3-methylcyclohexanol" as the filename, change the `_Save as type:_` drop down menu selection to `_MDL SD (*.sdf)_` and click `_Save_`. If a popup window appears with the title `_Select molecules_`, click `_Write all molecules_` followed by `_OK_`. Click `_OK_` in the popup window that confirms the file export. This step will export all structures from the conformational search to a single ".sdf" file—"cis-3-methylcyclohexanol.sdf"—located in the same directory as the conformational search output file. ****8 |****

`_Examine the resulting ".sdf" file to ensure that the results of the conformational search were correctly exported (Optional)_` Open the ".sdf" file in a text-editing application (e.g., TextEdit in Mac OS X or Notepad in Windows) and check that an entry is included for all unique conformations. A unique entry typically begins with the text "Spartan" followed by a series of numbers. Additionally, structures are usually systematically labeled, for example, the first conformation is by default titled "M0001". ****Create Gaussian input files for each conformer ● TIMING 15 min (Steps 9 – 12)**** ****9 |**** Download the "write-g09-inputs-sdf.py" (or "write-g03-inputs-sdf.py" if using Gaussian 03) Python script from ****Supplementary Data 1****:http://www.nature.com/protocolexchange/system/uploads/3133/original/Supplementary_Data_1.zip?1398228281 to the directory containing the ".sdf" file created in Step 7 (see Step 3 for directory location). If using Python version 2, download the "write-g09-inputs-sdf.py" script located in the Python-Version-2 directory within ****Supplementary Data 1****:http://www.nature.com/protocolexchange/system/uploads/3133/original/Supplementary_Data_1.zip?1398228281. ****▲ CRITICAL STEP**** Users must download the Python script from ****Supplementary Data 1****:http://www.nature.com/protocolexchange/system/uploads/3133/original/Supplementary_Data_1.zip?1398228281 that is compatible with the particular versions of both Python (i.e., either version 2 or 3) and Gaussian (i.e., either version 09 or 03) that are to be used. ****10 |**** `_Editing the "write-g09-inputs-sdf.py" Python script to change the memory and number of processors used in Gaussian calculations (optional)_` To accommodate different users' needs, we have prepared the "write-g09-inputs-sdf.py" Python script so that it is convenient to change the amount of memory and the number of the processors allocated to the computationally intensive Gaussian 09 jobs. Open the "write-g09-inputs-sdf.py" Python script in IDLE or any other Python script editor. Adjust the amount of memory used in the Gaussian 09 optimization/frequency and NMR jobs by changing the number to the right of "%mem=" on line 86 and 113, respectively. Adjust the number of core processors used in the Gaussian 09 optimization/frequency and NMR jobs by changing the number to the right of "%nproc=" on line 87 and 114, respectively. Save the edited script file in the same directory as the ".sdf" file created in Step 9. ****11 |**** In a command line interface application (i.e. Terminal for Mac OS X or Linux or Command Prompt for Windows) navigate to the directory that contains the ".sdf" file, the "write-g09-inputs-sdf.py" Python script and the associated computational files. ****12 |**** Execute the "write-g09-inputs-sdf.py" Python script (or the edited script that may have been created in Step 10 by entering the following command: `_> python write-g09-inputs-sdf.py cis-3-methylcyclohexanol.sdf_` The script will request the name of the candidate structure by displaying the following prompt: `_Enter the name of the candidate structure:_` Enter "cis-3-methylcyclohexanol" as the candidate structure name. Avoid using spaces when entering the name of the candidate structure. If the script executes successfully, the following message will be displayed: `_The script successfully performed the task of creating_ _Gaussian input files for each unique structure within the_ _cis-3-methylcyclohexanol.sdf file and moved these input files to the_ _cis-3-methylcyclohexanol-gaussian_files directory._` For each unique conformation within the associated ".sdf" file, the script will create two Gaussian input files. The script also creates a new directory labeled "cis-3-methylcyclohexanol-gaussian_files" and moves all of the Gaussian input files into this newly created directory. The Gaussian input files labeled "cis-3-

methylcyclohexanol-opt_freq-conf-#.com” are the input files for geometry optimization and frequency calculation. The Gaussian input files labeled “cis-3-methylcyclohexanol-nmr-conf-#.com” are the input files for NMR shielding tensor calculations. **?? TROUBLESHOOTING??** **Perform DFT calculations in Gaussian 09** (cf. Procedure in ref. 7) ● **TIMING 1 h** (Step 13) **13** | **To obtain the computed NMR data for the candidate structure, consult the Procedure in ref. 7 for instructions on using the resulting input files from Step 12 within Gaussian 09 to calculate** (i) DFT-optimized conformer geometries, (ii) free energies using the “opt_freq-conf” input files, and (iii) NMR shielding tensors using the “nmr-conf” input files. Additionally, the Procedure in ref. 7 includes Python scripts and detailed instructions for (i) assembling the free energy and NMR shielding tensor data into well-organized spreadsheet files, (ii) referencing and scaling the NMR data, (iii) determining the Boltzmann weighting factors of all conformers, and (iv) applying these weighting factors to generate the set of Boltzmann-weighted chemical shifts for the candidate structure. Details with regard to the choice of computational methodology (e.g., DFT functional and basis set preferences) are discussed in ref. 7. Additionally, the previously reported protocol⁷ highlights methods for determining the “best fit” for a candidate structure when comparing experimental spectral data to the computed NMR chemical shifts. Alternative approaches to determining the “best fit” have recently been reported by Goodman^{10,11} and Sarotti^{12,13}, and more traditional approaches are described in several excellent reviews^{14,15}.

Timing

A novice user can complete the Procedure described above in less than one hour. The time required to complete the molecular mechanics conformational search will increase with molecular complexity. However, in our experience this increase has not been substantial. Subsequent Gaussian computations will require significantly more computational time to complete, but the amount of active effort by the user is minimized because several steps have been automated with the use of Python scripts. A summary of the time required to complete various steps in the Procedure is shown below. Steps 1–4: <10 min of active effort; ca. 1–30 minutes to complete the conformational search depending on the structural complexity of the candidate structure. Steps 5–8: 15 min Steps 9–12: 15 min Step 13: ca. <60 min for the 3-methylcyclohexanols; timing depends on the number of conformational isomers and the structural complexity of the candidate structure.

Troubleshooting

See **Supplementary**

Table: http://www.nature.com/protocolexchange/system/uploads/3141/original/Troubleshooting_Table.pdf?1398228281 for Troubleshooting.

Anticipated Results

Following successful completion of the steps of the Procedure, six conformations of _cis_-3-methylcyclohexanol will be generated from the conformational search in Spartan, and the structure coordinates for each conformation will be exported to a “.sdf” file. Following execution of the Python script, “write-g09-inputs-sdf.py”, the directory “cis-3-methylcyclohexanol-gaussian_files” will be created, which will contain two Gaussian 09 input files for each conformation of the candidate structure. Once submitted to Gaussian 09, the input files having “opt_freq” in their title will instruct Gaussian to perform a geometry optimization and

frequency calculation of the included structural coordinates. Additionally, the input files having “nmr” in their title will instruct Gaussian to calculate NMR shielding tensors of the optimized geometry. For reference, we have provided the Spartan conformational search files and the “.sdf” file as “**Supplementary Data 2**”:http://www.nature.com/protocolexchange/system/uploads/3135/original/Supplementary_Data_2.zip?1398228537 and “**Supplementary Data 3**”:http://www.nature.com/protocolexchange/system/uploads/3137/original/Supplementary_Data_3.zip?1398228281, respectively. Additionally, the Gaussian 09 input files resulting from use of the Python script are included in “**Supplementary Data 4**”:http://www.nature.com/protocolexchange/system/uploads/3139/original/Supplementary_Data_4.zip?1398228281.

References

1. Nicolaou, K. C. & Snyder, S. A. Chasing molecules that were never there: misassigned natural products and the role of chemical synthesis in modern structure elucidation. *Angew. Chem. Int. Ed.* **44**, 1012–1044 (2005).
2. Hehre, W. J. A guide to molecular mechanics and quantum chemical calculations. Wavefunction, Inc., Irvine, CA, 2003.
3. Shao, Y., Molnar, L. F., Jung, Y., Kussmann, J. R., Ochsenfeld, C., Brown, S. T., Gilbert, A. T. B., Slipchenko, L. V., Levchenko, S. V., O'Neill, D. P., DiStasio, R. A., Jr, Lochan, R. C., Wang, T., Beran, G. J. O., Besley, N. A., Herbert, J. M., Yeh Lin, C., Van Voorhis, T., Hung Chien, S., Sodt, A., Steele, R. P., Rassolov, V. A., Maslen, P. E., Korambath, P. P., Adamson, R. D., Austin, B., Baker, J., Byrd, E. F. C., Dachsel, H., Doerksen, R. J., Dreuw, A., Dunietz, B. D., Dutoi, A. D., Furlani, T. R., Gwaltney, S. R., Heyden, A., Hirata, S., Hsu, C.-P., Kedziora, G., Khalliulin, R. Z., Klunzinger, P., Lee, A. M., Lee, M. S., Liang, W., Lotan, I., Nair, N., Peters, B., Proynov, E. I., Pieniazek, P. A., Min Rhee, Y., Ritchie, J., Rosta, E., David Sherrill, C., Simmonett, A. C., Subotnik, J. E., Lee Woodcock, H., III, Zhang, W., Bell, A. T., Chakraborty, A. K., Chipman, D. M., Keil, F. J., Warshel, A., Hehre, W. J., Schaefer, H. F., III, Kong, J., Krylov, A. I., Gill, P. M. W. & Head-Gordon, M. Advances in methods and algorithms in a modern quantum chemistry program package. *Phys. Chem. Chem. Phys.* **8**, 3172 (2006).
4. Gaussian 09, Revision A, Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., Scalmani, G., Barone, V., Mennucci, B., Petersson, G. A., Nakatsuji, H., Caricato, M., Li, X., Hratchian, H. P., Izmaylov, A. F., Bloino, J., Zheng, G., Sonnenberg, J. L., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Vreven, T., Montgomery, J. A., Jr, Peralta, J. E., Ogliaro, F., Bearpark, M., Heyd, J. J., Brothers, E., Kudin, K. N., Staroverov, V. N., Kobayashi, R., Normand, J., Raghavachari, K., Rendell, A., Burant, J. C., Iyengar, S. S., Tomasi, J., Cossi, M., Rega, N., Millam, N. J., Klene, M., Knox, J. E., Cross, J. B., Bakken, V., Adamo, C., Jaramillo, J., Gomperts, R., Stratmann, R. E., Yazyev, O., Austin, A. J., Cammi, R., Pomelli, C., Ochterski, J. W., Martin, R. L., Morokuma, K., Zakrzewski, V. G., Voth, G. A., Salvador, P., Dannenberg, J. J., Dapprich, S., Daniels, A. D., Farkas, Ö., Foresman, J. B., Ortiz, J. V., Cioslowski, J., Fox, D. J. Gaussian, Inc., Wallingford CT, 2009.
5. Jaguar, version 8.0. <http://www.schrodinger.com/citations/41/7/1/> (Schrödinger, LLC, New York, NY, 2013).
6. Bochevarov, A. D., Harder, E., Hughes, T. F., Greenwood, J. R., Braden, D. A., Philipp, D. M., Rinaldo, D., Halls, M. D., Zhang, J., Friesner, R. A. Jaguar: a high-performance quantum chemistry software program with strengths in life and materials sciences. *Int. J. Quantum Chem.* **113**, 2110–2142 (2013).
7. Willoughby, P. H., Jansma, M. J. & Hoye, T. R. A guide to small-molecule structure assignment through computation of (¹H and ¹³C) NMR chemical shifts. *Nature Protocols* **9**, 643–660 (2014).
8. MacroModel, version 10.0. <http://www.schrodinger.com/citations/41/11/1/> (Schrödinger, LLC, New York, NY, 2013).
9. ChemBio3D Ultra 13.0 Suite. http://www.cambridgesoft.com/Ensemble_for_Biology/ChemBio3D/
10. Smith, S. G. & Goodman, J.

M. Assigning the stereochemistry of pairs of diastereoisomers using GIAO NMR shift calculation. *J. Org. Chem.* **74**, 4597–4607 (2009). 11. Smith, S. G. & Goodman, J. M. Assigning stereochemistry to single diastereoisomers by GIAO NMR calculation: the DP4 probability. *J. Am. Chem. Soc.* **132**, 12946–12959 (2010). 12. Sarotti, A. M. & Pellegrinet, S. C. A multi-standard approach for GIAO ^{13}C NMR calculations. *J. Org. Chem.* **74**, 7254–7260 (2009). 13. Sarotti, A. M. Successful combination of computationally inexpensive GIAO ^{13}C NMR calculations and artificial neural network pattern recognition: a new strategy for simple and rapid detection of structural misassignments. *Org. & Biomol. Chem.* **11**, 4847–4859 (2013). 14. Lodewyk, M. W., Siebert, M. R. & Tantillo, D. J. Computational prediction of ^1H and ^{13}C chemical shifts: a useful tool for natural product, mechanistic, and synthetic organic chemistry. *Chem. Rev.* **112**, 1839–1862 (2012). 15. Tantillo, D. J. Walking in the woods with quantum chemistry—applications of quantum chemical calculations in natural products research. *Nat. Prod. Rep.* **30**, 1079–1086 (2013).

Acknowledgements

S.A.R. thanks Great Lakes Higher Education Guaranty Corporation and the Ripon College Center for Social Responsibility for funding. Additionally, we thank Thomas R. Hoyer for helpful input and Joseph D. Scanlon, Jordan Buhle, and Michael Enright for feedback during manuscript preparation.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [TroubleshootingTable.pdf](#)
- [SupplementaryData1.zip](#)
- [SupplementaryData2.zip](#)
- [SupplementaryData3.zip](#)
- [SupplementaryData4.zip](#)