

ARGO

Users Manual
Version 1.3



August 24, 2023

E. C. Semidalas and C. E. Semidalas
email: msemidalas@yahoo.com
chsemid@uniwa.gr

1 Introduction

The ‘Argo’ program draws its name from ancient Greek, referencing the vessel of the same name commissioned for Jason’s renowned Argonauts expedition. This choice symbolizes the pursuit of making the best use of quantum chemical calculation results.

ARGO facilitates the analysis of results obtained from quantum chemistry codes, specifically Gaussian.[1] This task is achieved through a set of Python scripts and the following capabilities are currently available:

- Visualize potential energy surfaces (PES) of molecules.
- Calculate Raman intensities from Raman activities and plot bar graphs of IR and Raman intensities.

Reference

Kindly cite this publication when acknowledging ARGO[4] in your work:

Semidalas EC, Semidalas CE. Argo: a data analysis program for quantum chemical calculations. J Mol Model. 2019;25(3):82. doi:10.1007/s00894-019-3975-x

The program ARGO is provided AS IS with NO WARRANTY OF ANY KIND. It may be freely distributed and modified.

Contents

1	Introduction	1
2	Running ARGO	3
3	Argo scripts	3
3.1	Visualizing potential energy surfaces	4
3.2	IR and Raman intensities	6

Version Updates

- The manual underwent a comprehensive rewrite.
- The code was improved to ensure compatibility with Python 3.8+ and Gnuplot 5.4+.
- ARGO generates bar graphs depicting IR and Raman intensities.

2 Running ARGO

ARGO is a program developed in Python programming language for the analysis of results from electronic structure calculations with the Gaussian program. It has been tested under macOS Ventura 13.4.1 and Linux openSUSE Leap. The program's functionalities include data processing and the construction of Potential Energy Surface (PES) graphs using Gnuplot.

System Requirements:

1. Python 3.8.2 or later
2. Gnuplot 5.4 or later
3. The Python pandas, numpy, and matplotlib libraries are required.

ARGO can be downloaded from these two links at Sourceforge and Github:

- <https://sourceforge.net/projects/argo1/files/Argo%201.3/>
- <https://github.com/msemidalas/ARGO>

Save the zipped folder and extract the files. The Python scripts are located in the folder 'source'.

3 Argo scripts

Overview

A brief description of the scripts provided in ARGO can be found below

Script	Utility
PESVisualizer.py	Collects dihedral angles and energies from a potential energy surface scan and generates 2D and 3D plots.
IR_Raman_info.py	Gathers frequencies, IR and Raman Intensities, and Raman activities. Transforms Raman activities to Raman intensities and plots them in bar graphs.

Limitations

The `PESVisualizer.py` is restricted to PES scans involving two variables only. *Do not use* this script for single or geometric scans involving more than two parameters.

3.1 Visualizing potential energy surfaces

The script `PESVisualizer.py` processes data from Gaussian output files and generates plots of potential energy surfaces (PES) using Gnuplot. It performs several tasks such as extracting data from the output file, processing it into CSV files, and creating Gnuplot scripts for 3D and 2D PES plots.

Save the script in a directory where you have your Gaussian output file(s) and where you want to store the generated plots.

Open a terminal or command prompt and run the script:

```
1 python3 PESVisualizer.py
```

Enter the full name of your Gaussian output file. For example, consider the `ht_PES.log` file for the PES scan of the hydroxytyrosol (HT) molecule.

```
1 Enter the full name of your Gaussian output file:  
2 ht_PES.log
```

It will then automatically search for specific start and end strings within the output file to identify the relevant data section.

Enter the scanned geometrical parameters, e.g., the dihedral angles are D13 and D16 in our example:

```
1 Enter dihedral angle 1:  
2 D13  
3 Enter dihedral angle 2:  
4 D16  
5 Do you want to delete intermediate files? (yes or no)  
6 yes
```

All data are saved in the file 'final.dat'. Also, instructions are generated for creating 2D and 3D-PES plots in Gnuplot. To do so, run:

```
1 gnuplot 2dPES.txt  
2 gnuplot 3dPES.txt
```

Note: The x and y-axis ranges in the plots may need adjustment to suit your particular scans. The same applies to the grid settings. In the case of our example for HT, we opted for 19x19 grid points as they aligned with the scan steps (361 conformers here).

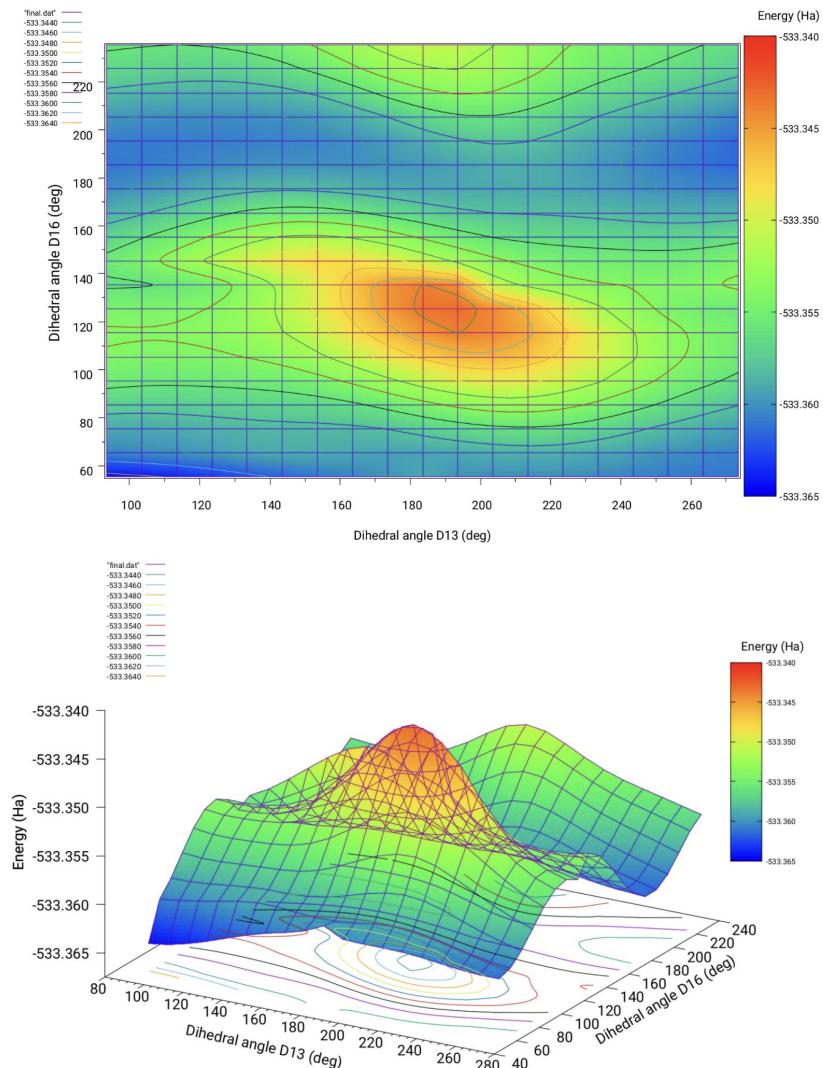


Figure 1: 2D and 3D-PES plots of hydroxytyrosol

3.2 IR and Raman intensities

To extract the frequencies, IR and Raman intensities and activities from the Gaussian output file, use the IR_Raman_Info.py script. Here, we provide an example for the methanol molecule.

- Run the IR_Raman_Info.py script in the directory where your Gaussian output file is located in by typing:

```
1 python3 IR_Raman_Info.py
```

- You will be required to provide the output filename, laser beam frequency (in cm^{-1}), and temperature (in K):

```
1 Enter the filename (e.g., ch3oh.log):
2 ch3oh.log
3 Insert frequency value of laser excitation (cm-1):
4 18796.99
5 Insert temperature value (K): (e.g., 298.15 K)
6 298.15
```

The end result is a file called IR_Raman.csv containing data on IR and Raman intensities and activities.

Freq. (cm^{-1})	IR intensity (km/mol)	Raman	
		activity ($\text{\AA}^4/\text{amu}$)	Raman intensity
ω_1	538.92	123.05	2.98
ω_2	1114.81	83.97	9.46
ω_3	1169.13	39.82	1.39
ω_4	1246.56	2.87	4.33
ω_5	1514.99	46.95	1.58
ω_6	1582.79	10.85	2.98
ω_7	1597.55	2.00	14.16
ω_8	1606.70	2.98	14.36
ω_9	3169.82	57.01	144.11
ω_{10}	3230.85	87.20	73.93
ω_{11}	3291.97	44.34	80.85
ω_{12}	3976.59	46.60	69.05

The Raman activities are converted to Raman intensities through the following equation:[2, 3]

$$I_i = \frac{(2\pi)^4}{45} (v_0 - v_i)^4 \frac{h}{8\pi^2 c v_i (1 - \exp(-\frac{hv_i c}{kT}))} S_i \quad (1)$$

Here, S_i , v_i , v_0 , h, c, k, and T represent Raman activities, frequency of the i^{th} band, frequency of the incident laser, Planck constant, speed of light, Boltzmann constant, and temperature, respectively.

In addition to that, the spectral data can be plotted in two bar graphs.

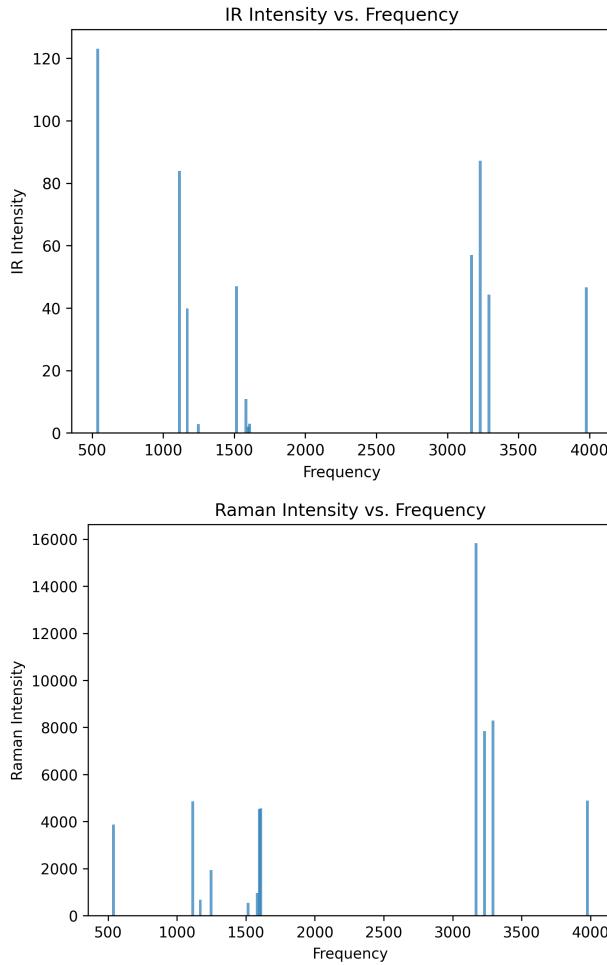


Figure 2: IR and Raman intensities for the methanol molecule

References

- [1] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Jamesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zatkowski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, Jr. Montgomery, J. A., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, C Tomasi, J., M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J Fox. Gaussian 16, Rev. C.01, 2016. ISSN 1424-859X.
- [2] G. Keresztury, S. Holly, G. Besenyei, J. Varga, Aiying Wang, and J.R. Durig. Vibrational spectra of monothiocarbamates-II. IR and Raman spectra, vibrational assignment, conformational analysis and ab initio calculations of S-methyl-N,N-dimethylthiocarbamate. *Spectrochimica Acta Part A: Molecular Spectroscopy*, 49(13-14):2007–2026, nov 1993. ISSN 05848539. doi: 10.1016/S0584-8539(09)91012-1. URL <https://linkinghub.elsevier.com/retrieve/pii/S0584853909910121>.
- [3] P L Polavarapu. Ab initio vibrational Raman and Raman optical activity spectra. *The Journal of Physical Chemistry*, 94(21):8106–8112, oct 1990. ISSN 0022-3654. doi: 10.1021/j100384a024. URL <https://doi.org/10.1021/j100384a024><https://pubs.acs.org/doi/abs/10.1021/j100384a024>.
- [4] E. C. Semidalas and C. E. Semidalas. Argo: a data analysis program for quantum chemical calculations. *Journal of Molecular Modeling*, 25(3):82, 2019. ISSN 1610-2940. doi: 10.1007/s00894-019-3975-x. URL <http://link.springer.com/10.1007/s00894-019-3975-x>.