

A Code Glitch May Have Caused Errors In More Than 100 Published Studies

The discovery is a reminder that science is collaborative and ideally self-correcting, but that nothing can be taken for granted.

By Maddie Bender - Oct 10 2019



Scientists in Hawai‘i have uncovered a glitch in a piece of code that could have yielded incorrect results in over 100 published studies that cited the original paper.

The glitch caused results of a common chemistry computation to vary depending on the operating system used, causing discrepancies among Mac, Windows, and Linux systems. The researchers published the revelation and a debugged version of the script, which amounts to roughly 1,000 lines of code, on [Tuesday in the journal Organic Letters](#).

“This simple glitch in the original script calls into question the conclusions of a significant number of papers on a wide range of topics in a way that cannot be easily resolved from published information because the operating system is rarely mentioned,” the new paper reads. “Authors who used these scripts should certainly double-check their results and any relevant conclusions using the modified scripts in the [supplementary information].”

Yuheng Luo, a graduate student at the University of Hawai‘i at Mānoa, discovered the glitch this summer when he was verifying the results of research conducted by chemistry professor Philip Williams on cyanobacteria. The aim of the project was to “try to find compounds that are effective against cancer,” Williams said.

Under supervision of University of Hawai‘i at Mānoa assistant chemistry professor Rui Sun, Luo used a script written in Python that was published as part of a 2014 paper by Patrick Willoughby, Matthew Jansma, and Thomas Hoye in the journal Nature Protocols. The code computes chemical shift values for NMR, or nuclear magnetic resonance spectroscopy, a common technique used by chemists to determine the molecular make-up of a sample.

Luo’s results did not match up with the NMR values that Williams’ group had previously calculated, and according to Sun, when his students ran the code on their computers, they realized that different operating systems were producing different results. Sun then adjusted the code to fix the glitch, which had to do with how different operating systems sort files.

Willoughby, the first author of the 2014 study who wrote the script, called the new study “a beautiful example of science working to advance the work we reported in 2014.”

“They did a tremendous service to the community in figuring this out,” he said.

Williams said that the original study is an elegantly written paper that is “incredibly useful to a large group of people, and the error is very subtle.” Nevertheless, the researchers believe that the glitch could have produced serious downstream effects.

For example, if the code led Williams to wrongly identify the contents of his sample, chemists trying to recreate the molecule to test as a potential cancer drug would be chasing after the wrong compound, Williams said.

“The process of science here worked exactly as it’s supposed to”

It is unclear how many papers this glitch may have affected - researchers do not typically disclose the operating system they use for their analyses since it should be irrelevant, Williams said. According to metrics from Nature Protocols, the 2014 paper has been accessed nearly 1,900 times and cited in 158 other studies. However, not every study that cited the paper may have used the script.

Rob Keyzers, a chemistry lecturer at Victoria University of Wellington in New Zealand who had cited the protocol in a study published this year, said in an email that he was not aware of the glitch. He added that he was not “unduly worried” about his results since his group did not use the script containing the glitch. “I will certainly check our data carefully to make sure that we are not making any undue claims though,” he said.

In late July, Williams and Sun reached out to the authors of the original paper, alerting them to the glitch. Williams said he hoped that by working together, they could bring the problem to the attention of researchers who had used the code.

“I personally haven’t had the opportunity to go to the original authors [of a study] and announce a bug, so I wasn’t sure what to expect,” Sun said, adding that the authors were “very gracious” and encouraged him and Williams to publish their findings.

“The process of science here worked exactly as it’s supposed to,” Williams said.

Process chemist Lucas Moore, who tweeted about the study, said in an email that “news of a bug like this is something that we in the scientific community tend to spread far and wide—we really want to get it right.”

Willoughby and Hoye plan to update the Nature Protocols study acknowledging the glitch and providing Sun’s fixed code, according to correspondence from Hoye quoted in the new paper.

In a statement, a spokesperson for Nature Protocols said that they are looking into the issues raised in the new study, but for confidentiality reasons cannot comment on individual cases.

The incident is a reminder that science is collaborative and ideally self-correcting, but that nothing can be taken for granted.

“This is a very small and subtle error,” Williams said. “We all kind of assume that a computer program always spits out the correct answer.”

[Article Link](#)

OL | Organic Letters | Letter
Cite This: *Org. Lett.* XXXX, XXX, XXX–XXX | pubs.acs.org/OrgLett

Characterization of Leptazolines A–D, Polar Oxazolines from the Cyanobacterium *Leptolyngbya* sp., Reveals a Glitch with the “Willoughby–Hoye” Scripts for Calculating NMR Chemical Shifts

Jayanti Bhandari Neupane, Ram P. Neupane, Yuheng Luo, Wesley Y. Yoshida, Rui Sun, and Philip G. Williams*

Department of Chemistry, University of Hawai‘i at Mānoa, 2545 McCarthy Mall, Honolulu, Hawaii 96822, United States

Supporting Information

ABSTRACT: The bioactivity-guided examination of a *Leptolyngbya* sp. led to the isolation of leptazolines A–D (1–4), from the culture media, along with two degradation products (5 and 6). Density functional theory nuclear magnetic resonance calculations established the relative configurations of 1 and 2 and revealed that the calculated shifts depended on the operating system when using the “Willoughby–Hoye” Python scripts to streamline the processing of the output files, a previously unrecognized flaw that could lead to incorrect conclusions.

“Willoughby–Hoye” Scripts from 2014 Nature Protocols

Ubuntu16	δ_{C1} 172.4 (Incorrect)
Windows10	δ_{C1} 173.2
Mac Mavericks	δ_{C1} 173.2
Mac Mojave	δ_{C1} 172.7 (Incorrect)

Different Calculated Chemical Shifts!

In theory, the chemical shift of each atom from each isomer should be properly weighted according to the free energy of the conformer, but this was not consistently the case. In the end, the inconsistency was traced to differences in the default file-sorting algorithm in Python across platforms, as shown in Figure 2, and the fact that, as written, the script “**nmr-data_compilation**” assumes that the frequency and NMR files are sorted in the same order. The Boltzmann contribution calculated from the first frequency file in the directory is simply applied to the chemical shifts calculated in the first NMR file. For example, in Windows 10, the script works as designed: Two groups of output files, ***opt_freq*-ID.out** containing free energies and ***nmr*-ID.out** containing chemical shifts, are automatically sorted by the system such that the conformers are paired correctly (Figure 2B).

However, in LINUX there is no default sorting of file names because this depends on the local settings, and the scripts, as designed, do not check that the frequency and the NMR files are properly matched. As a result, the free energies could be paired with chemical shifts from different conformers (Figure 2A), which leads to incorrectly calculated chemical shifts that could surely lead to a wrong final conclusion.

To overcome this issue, we amended the script to include a line in the **read_gaussian_outputfiles** subroutine that forces sorting before pairing and a longer file-matching check function that alerts the user when

there is a potential file-matching issue (see the Supporting Information). We have tested the revised script across platforms (i.e., MacOS, LINUX, and Windows) running different versions of Python, and it yields consistent results.

B) Windows Directory Listing

```
1b-opt_freq-conf-1.out 1b-nmr-conf-1.out
1b-opt_freq-conf-10.out 1b-nmr-conf-10.out
1b-opt_freq-conf-11.out 1b-nmr-conf-11.out
....
```

A) LINUX Directory Listing

```
1b-opt_freq-conf-16.out 1b-nmr-conf-10.out
1b-opt_freq-conf-15.out 1b-nmr-conf-5.out
1b-opt_freq-conf-11.out 1b-nmr-conf-8.out
....
```

Figure 2. Inconsistent sorting of files.

This simple glitch in the original script calls into question the conclusions of a significant number of papers on a wide range of topics in a way that cannot be easily resolved from published information because the operating system is rarely mentioned. In the first half of 2019 alone, the protocol was referenced/used during the elucidation of several natural products, to characterize reaction products and to understand biosynthetic pathways. Authors who used these scripts should certainly double-check their results and any relevant conclusions using the modified scripts in the SI. Ultimately, this example serves as a reminder of the principle Caveat emptor and that users should validate noncommercial software on their system prior to use on new applications.