How Open Is Commercial Scientific Software?

Most scientific research nowadays relies on some kind of software. This is particularly true in fields such as my own, quantum chemistry. Quantum chemical software is used in applications to study various problems in chemistry, often in close connection with experiment or by experimental groups themselves. In addition, it serves as a platform for the development of new theoretical and computational methods.

Many quantum chemical program packages are available that differ in functionality, usability (from easy-to-use by nonspecialists to usable only by the person who wrote it), and computational efficiency. What all available codes have in common is that, at least initially, they have been developed with public (i.e., tax payers') money. Nevertheless, the terms under which they are made available differ very significantly. Some program packages\(^1\) are available under open-source licenses (meaning that anyone can “study, change, and distribute the software to anyone and for any purpose\(^2\))\(^2\), while others are owned by commercial companies who sell them to both academic groups and users in industry for a small or large license fee. Intermediate models (free of charge but not open-source) also exist, such as closed-source software that is distributed free of charge to academic groups\(^3\) or software for which the source code is available to academic users but with license terms that prohibit changes or redistribution.\(^4\)

Pushing toward Open-Source in Science. Open-source scientific software offers a number of advantages for science as a whole. Numerous arguments have been made for the use of open-source licenses in academic software development. Many of these arguments have recently been summarized by Gezelter in a Viewpoint in this journal,\(^5\) and in an excellent blog post by Katz et al.,\(^6\) and I will refrain from repeating these arguments here. This has led some funding agencies to require software developed under certain grants to be released under open-source licenses. Intermediate models (free of charge but not open-source) also exist, such as closed-source software that is distributed free of charge to academic groups\(^3\) or software for which the source code is available to academic users but with license terms that prohibit changes or redistribution.\(^5\)

This Viewpoint is written by a group of eminent scientists, whose work in quantum chemical method and software development I admire. All of the authors of ref 8 have in common that, besides being professors at research universities, they are co-owners of companies selling quantum chemical software packages.\(^9\) I disagree with many of the arguments put forward by the authors. Instead of dissecting these arguments individually,\(^10\) I want to focus on one particular perspective here and outline some of the problems that closed-source academic software development cause for scientists working in quantum chemical software development, especially for early career scientists.

Perspective of the Method Developer. To develop, test, and finally use a new idea in quantum chemistry, it needs to be implemented in software. Usually, this requires the use of many well-established tools, such as integral codes, basic methods developed many decades ago, and advanced numerical algorithms. All of these are a prerequisite for new developments but not “interesting” by themselves anymore today. Even though all of these tools are well-documented in the scientific literature, recreating these “lower layers” would be a major effort that cannot be repeated every time and by every research group because both time and funding are limited resources. This is especially true for young researchers at the beginning of their scientific career, who have only small groups and are under pressure to deliver novel contributions within a limited time frame in order to have a chance to secure a permanent position.

Therefore, method developers in quantum chemistry need some existing program package as a “development platform”. Both open-source and commercial codes can offer such a platform. Open-source codes have the advantage that there is no barrier to access. Anyone can download the source code and start working on a new method. I have so far mostly contributed my developments to commercial codes. These also offer certain advantages. For successful codes, the revenues from selling licenses can be used by the companies owning them to employ software developers who maintain and document the code. These can further improve code contributed by academic groups in order to make it maintainable, efficient, and easily extendable. This can speed up new developments and improve the quality and efficiency of the resulting new software.

Commercial Codes as “Open Teamwork”? The authors of ref 8 argue that there is no need for open-source development platforms because many commercial codes, such as Q-Chem\(^11\) and others, operate under what they call an “open teamwork” model. As they point out, many commercial code have assembled rather large communities of academic developers.

However, access to commercial software packages as a development platform is not as open as the authors of ref 8 claim. First of all, it is subject to signing a developer agreement, the terms of which are dictated by the companies owning the source code and that are drafted to protect their commercial interests. Usually, they include a transfer of intellectual property rights for the new developments to these companies as well as nondisclosure clauses concerning the source code and algorithms implemented in it. Often, such developer agreements require exclusivity, meaning that new source code cannot be contributed to other commercial or open-source packages. Sometimes, developers are even banned from using competing program packages.\(^12\)

Such requirements for exclusivity prevent scientific collaborations. I have personally encountered this on several occasions, when fellow scientists told me that they would love to collaborate but that they cannot do so because we are competing in the same robust packages. Thus, the commercial interests of software companies lead to a segregation of the scientific community based on affiliation with certain codes. Often, methods developed in one program package are
reinvented in others because scientists cannot collaborate or use each others’ software.

More severely, the need to protect the intellectual property of commercial software packages has even lead to attempts to obtain patents on quantum chemical methods and algorithms.13 Such patents, if granted, would bar researchers affiliated with other program packages from pursuing method development in certain areas and would also prevent the development of open-source implementations of such methods.

Another problem faced by early career researchers contributing to commercial software packages is that when moving from one institution or research group to another, they may not be able to continue working on the software that they developed over several years and might even be cut off from using the software to which they contributed. Therefore, researchers may be forced to restart from scratch without many resources, which is often not affordable in the fast competing world of science. Thus, many will choose to stay within the same academic school throughout their career, which further preserves a segregation of the scientific community.

Perpetuating Power Structures. The use of commercial codes as development platform also puts the few scientists owning the corresponding companies into a gatekeeper position. It is up to them to decide who is allowed to contribute new ideas and developments. The policies of different companies may differ significantly. However, all of them will require revealing novel research ideas to the scientists in these gatekeeper positions. These will in many cases be competing scientists, who might reject access because ideas are opposite to their own "scientific beliefs" or because they might interfere with their own lines of research.

These mechanisms lead to perpetuating power structures that put very few individual scientists, the owners of commercial software packages, in control of most method development. It should be pointed out here that many of the authors of ref 8 are not the original developers of the commercial codes that they now own but that they have inherited these codes from their academic mentors. Such decisions of the few pioneers who developed the software infrastructure that our whole field relies on today will certainly have been based on scientific achievements, but they have not been taken by the academic community as a whole through peer review and funding panels.

This contradicts the merit-based access to scientific resources that the authors of ref 8 so keenly advertise. The possibility to carry out new method developments should only be based on the quality of new ideas and not on whether or not a scientist is part of a certain school or affiliated with a certain software package. The "track record of productivity"8 rewarded by funding agencies with grant money should have been established with competitive ideas not because of access to a software infrastructure built by a researcher’s academic ancestors. Again, let me point out that I admire the track record of all of the authors of ref 8, but I think that the playing field has to be leveled for the next generation of scientists.

Gatekeeping in Open-Source Software. At least in part, the problems discussed above also exist for open-source program packages. Because of the lack of revenue from selling licenses, these codes might be less well documented and maintained. This has the consequence that, even if there is no formal barrier, the technical barrier to contributing to them might be significant. Often, it can only be overcome by collaborating with one of the lead authors of such open-source program packages, which again puts these into a gatekeeper position. For academic program packages that are free of charge but not truly open source, even the formal barrier remains, and it might be necessary to collaborate with the lead authors in order to obtain the source code or permission to modify it. In this case, not commercial interest but the wish to retain this “scientific revenue” will keep the original authors from moving to true open-source licenses.

For the same reason, authors of open-source program packages often do not immediately release new developments to the public in order to maintain a competitive advantage. While this is often understandable from an individual scientist’s point of view, it prevents other scientists from immediately using new methods and from extending them with their own developments and thus delays the progress of the field as a whole. Requirements of journals to publicly release open-source software described in articles upon publication could alleviate this problem.

Possible Solutions. A first step toward a solution would be to clearly reveal and if possible remove the conflict of interest many scientists owning and running scientific software companies face. If these companies are run by businessmen instead of active scientists, then decisions to grant access to new external developers will be based on the possible merits for the paying users of the software packages and will not be influenced by fear of scientific competition. Some commercial codes, such as the Amsterdam Density Functional (ADF) package,15 use such a model. A least the policies underlying decisions whether or not to grant access to external developers should be made transparent. Nevertheless, while such models might alleviate the gatekeeping problem, other problems of closed-source scientific software will remain.

Second, I believe that funding initiatives aimed exclusively at open-source packages are an important piece in creating truly open platforms for method development. Such initiatives provide a means to level the playing field, by making funding available to open-source packages that commercial codes can obtain via their revenue stream form selling licenses. It is, however, important that they include funding not only for new scientific developments but also for sustaining the maintenance of open-source software. This must include the possibility to hire software engineers as nonscientific staff as well as training programs for students entering scientific software development. Such open-source software sustainability initiatives are now starting to be implemented in various countries, for example, in the U.K.,15 the U.S.,16 and, at least partly, in The Netherlands.17 Such initiatives should, of course, not destroy commercial codes but allow open-source software to coexist with commercial program packages. In fact, there are also funding opportunities that are exclusively available to commercial codes, such as technology grants. In Europe, many programs under the Horizon2020 framework encourage or require the involvement of small or medium enterprises, and some scientific software companies have been very successful in securing such grants.18

Concerning funding for fundamental research, open-source mandates might indeed have severe consequences for commercial codes because the mandates would cut them off from academic method development. This could be mitigated by requiring such codes, if they want to profit from public funding for basic research, to implement a truly open platform strategy that allows nondiscriminatory access to the source code for interested developers. With strict open-source mandates, commercial codes would still have the possibility
to create new development in the form of modular libraries released under open-source licenses. Many academic contributors to commercial program packages already choose to do so.\(^\text{19,20}\) Finally, it should be pointed out that many successful business models exist for commercializing open-source software,\(^\text{21}\) without interfering with the benefits of open-source licenses for science as a whole.

**Christoph R. Jacob*\(^\text{2}\)**

Institute of Physical and Theoretical Chemistry, TU Braunschweig, Hans-Sommer-Straße 10, 38106 Braunschweig, Germany

* AUTHOR INFORMATION

**Corresponding Author**

*E-mail: c.jacob@tu-braunschweig.de. Twitter: @chjacob.

**Notes**

Views expressed in this Viewpoint are those of the author and not necessarily the views of the ACS.

The author declares the following competing financial interest(s): Most of my past method development has been contributed to the commercial software package ADF, owned by Scientific Computing & Modelling N.V., Amsterdam, under a developer agreement. I have no financial stakes in SCM or other scientific software companies, and I did not receive direct or indirect financial compensation for these contributions. I have also contributed to the Dirac and Dalton packages, which are free for academic users, but not (yet) open source. Some software developed in my research group is - or will soon be - available under open-source licenses.

This Viewpoint is based on a blog post by the author that first appeared at URL http://blog.christophjacobs.eu/2015/07/18/how-open-are-commercial-scientific-software-packages/.

**ACKNOWLEDGMENTS**

The author thanks Alan Aspuru-Guzik, Radovan Bast, T. Daniel Crawford, J. Daniel Gezelter, Andre Severo Pereira Gomes, Vedran Miletic, Michele Pavanello, Judith Rommel, and Lucas Visscher for critically commenting on the manuscript and in addition thanks T. Daniel Crawford, J. Daniel Gezelter, and Peter Schwerdtfeger for encouragement.

**REFERENCES**


(9) See the Conflict of interest statement at the end of ref 8.


