analytical chemistry

Perspective

pubs.acs.org/ac

Upon the Shoulders of Giants: Open-Source Hardware and Software in Analytical Chemistry

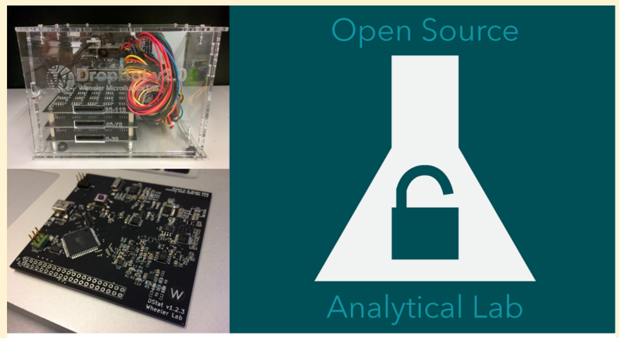
Michael D. M. Dryden,[†] Ryan Fobel,[†,‡] Christian Fobel,[†,‡] and Aaron R. Wheeler*[,†,¶,‡]

[†]Department of Chemistry, University of Toronto, 80 Saint George Street, Toronto, Ontario M5S 3H6, Canada

[‡]Donnelly Centre for Cellular and Biomolecular Research, 160 College Street, Toronto, Ontario M5S 3E1, Canada

[¶]Institute of Biomaterials and Biomedical Engineering, University of Toronto, 164 College Street, Toronto, Ontario M5S 3G9, Canada

ABSTRACT: Isaac Newton famously observed that "if I have seen further it is by standing on the shoulders of giants." We propose that this sentiment is a powerful motivation for the "open-source" movement in scientific research, in which creators provide everything needed to replicate a given project online, as well as providing explicit permission for users to use, improve, and share it with others. Here, we write to introduce analytical chemists who are new to the open-source movement to best practices and concepts in this area and to survey the state of open-source research in analytical chemistry. We conclude by considering two examples of open-source projects from our own research group, with the hope that a description of the process, motivations, and results will provide a convincing argument about the benefits that this movement brings to both creators and users.



The enterprise of scientific research is succinctly described by a quote made famous by Sir Isaac Newton: "If I have seen further it is by standing on the shoulders of giants"—that is, any given scientific advance represents the sum of activities that are themselves built on the foundation of past efforts by other researchers. Thus, the advancement of science depends on discovering, accessing, and building upon the research of others, tasks which are facilitated in our current world via the Internet. With this in mind, scientists are increasingly choosing to publish their research in open-access journals, including American Chemical Society (ACS) titles, such as *ACS Central Science* and *ACS Omega*, and journals from many other publishers, including the Public Library of Science (PLOS) and eLife Sciences. Furthermore, a growing number of researchers are supplementing traditional publication by submitting to preprint archives, including the well-known arXiv (physics, mathematics, computer science; over 1.2 million articles since 1991, 10 000+ submitted monthly),[1] as well as the newer bioRxiv (biology),[2] and the recently announced ACS chemRxiv (chemistry).[3] Perhaps motivated by Newton's broad shoulders, authors are gravitating toward these mechanisms, recognizing that (as indicated in the *ACS Omega* "Vision Statement") open-source publications make science "accessible beyond the traditional academic readership," including "readers in industry, at policy institutions, the media, and the general public."[4]

While open-source publishing is making communication about science more accessible, there is often a divide between the content of a publication and the reproducibility of the methods described therein—that is, in many publications, technical details that are required for others to repeat the work are missing. Reproducibility is especially important in publications that describe custom hardware or software, which may be challenging for others to use even when the authors have made good-faith efforts to provide all of the details necessary for reproduction. Although this problem abounds in many disciplines (as well as in many subdisciplines of Chemistry), "analytical chemistry" in particular is often associated with the creation of new instruments and apparatus as well as software to control them. For example, a recent issue of this journal (*Analytical Chemistry*, Issue 2, Volume 89) included 25 publications that described custom hardware or software, out of 48 contributions. While these numbers make it clear that analytical chemists are enthusiastic about developing custom hardware and software systems, of these 25 contributions, the majority lack important details (including printed circuit board designs and software source code), which preclude them from being fully repeatable by other authors. It is of course possible to recreate hardware and software from descriptions in published work, but in practice, this approach at best leads to duplicated effort and at worst may prevent replication and reuse because of unstated assumptions and missing design details. In recognition of these challenges, we

Received: February 8, 2017
Accepted: March 27, 2017
Published: April 5, 2017

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338

advocate the increased adoption of open hardware and software principles in analytical chemistry research.

"Open-source" software, which has been around for decades, is also commonly referred to as Free and Open-Source Software (FOSS). In this case, "free" refers to freedom, as FOSS is software for which the source code is not only freely available but is explicitly licensed to allow users to use and modify it as they wish. The most obvious success of open-source software is its ubiquity: FOSS can be found nearly everywhere, from Android phones to the web servers used by over 70% of the world's million busiest websites (many of which also run the open-source Linux operating system).[5] There is also great precedent for chemists releasing software as FOSS: for example, molecular editing and visualization software,[6] toolkits for manipulating chemical data formats,[7] and a plethora of quantum chemistry packages are available and widely used.[8−10] Interest in open-source software in analytical chemistry has increased significantly since the turn of the century; though as shown in Figure 1, the percentage of publications describing or mentioning open-source software is still modest. For example, the issue of *Analytical Chemistry* mentioned above contained two reports of software released as FOSS.[11,12]

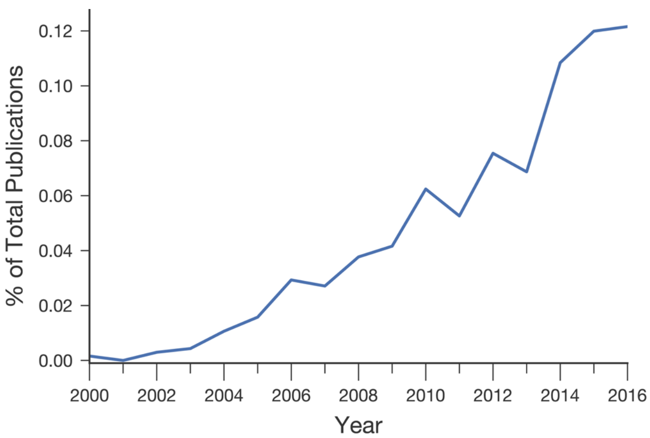


Figure 1. Scopus results from 2000−2016 for "open source" and "analytical chemistry" (including some related terms) excluding results from subject areas outside physical or natural sciences, normalized by the same search not including "open source."

While open-source software is widely known, a similar concept for hardware exists, called Open-Source Hardware (OSHW), which is simply an instrument for which the "design is made publicly available so that anyone can study, modify, distribute, make, and sell the design or hardware based on that design."[13] OSHW has recently been applied specifically to scientific tools by the Global Open Science Hardware movement (GOSH), with the stated goal to "reduce barriers between diverse creators and users of scientific tools to support the pursuit and growth of knowledge."[14] The field with the strongest adoption of OSHW and GOSH principles is experimental physics; for example, the Open Hardware Repository operated by the European Council for Nuclear Research (CERN) lists over one hundred projects from radiation-tolerant electronic modules to cosmic ray detectors, many of which have hundreds or thousands of units deployed in CERN facilities.[15] Scientists working in the life sciences have also created a growing list of open-source tools, covering areas including microscopy, molecular biology, and electrophysiology.[16] Likewise, the US National Institutes of Health (NIH) "3D Print Exchange" includes designs for hardware related to health sciences,[17] and a number of open-source tools from many fields have been curated in a PLOS collection.[18] Conversely, analytical chemistry has not experienced the same penetration of ideas like OSHW and GOSH. Although there are certainly some open-source analytical tools ranging from photometric measurement systems[19] and quartz crystal microbalances[20] to more mundane tools like tube racks,[21] centrifuges,[22] and well plates and micropipettes,[23] publishing open-source hardware does not seem to be front-of-mind for many analytical chemists.

This Perspective is not the first to address this topic; other authors, in particular Joshua Pearce with his excellent *Open-Source Lab* textbook,[24] have made strong arguments for wider adoption of the open-source philosophy in science, including the reduced cost compared to commercial products, the practice of in-depth peer-review on a massive scale, expanded visibility and impact, and simply improving the efficiency of scientific communication. These arguments are particularly apt for analytical chemistry because of the importance of instrumentation in this field, but there are no contributions in the literature (known to us) with an explicit focus on how open-source concepts intersect with analytical chemistry research. Thus, we write to introduce analytical chemists who are new the open-source movement to best practices and concepts in this area, and to survey the state of open-source research in analytical chemistry. We conclude with discussions of two of our lab's open-source hardware projects: DropBot, an advanced digital microfluidic droplet handling system,[25] and DStat, a versatile, compact potentiostat.[26] Through these examples, we hope to describe the process of developing and releasing open-source projects, as well as to make a convincing argument about the benefits this brings their creators and users.

■ OPEN SOURCE ADOPTION IN ANALYTICAL CHEMISTRY

Generic Components and Tools. A key aspect of the open-source movement in analytical chemistry has been the development of components and tools that are useful in creating new laboratory instrumentation. The most obvious success-stories in this category are Arduino and RepRap. Arduino is the name given to a combination of an open-source software framework for simplifying the programming of microcontrollers with a series of general-purpose OSHW boards bearing Atmel microcontrollers, creating a simple and inexpensive platform for prototyping electronics.[27] Analytical chemists have made use of the Arduino system for diverse purposes (including data acquisition and instrument control) for techniques including electrochemical[28,29] and optical detection,[30] mass spectrometry,[31] separations,[32,33] and microfluidics.[25,34,35]

RepRap stands for Replicating Rapid-prototyper, a name given to a series of open-source 3D printers designed with the goal of being self-replicating (i.e., a large portion of a RepRap's components can be printed with a RepRap).[36] The RepRap project has led the transition of 3D printers from large commercial/industrial instruments costing tens of thousands of dollars to benchtop units easily accessible by academic laboratories and hobbyists. With a few exceptions, RepRap printers operate based on the fused filament fabrication technique, where plastic parts are created by stacking 2D layers of molten plastic dispensed by a heated nozzle on an XYZ

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338

stage, gradually producing a 3D object. Though RepRaps are generally unsuitable for printing the very fine features required to produce microfluidic devices (though examples of RepRap-printed large channel devices exist[37]), they are useful for rapid prototyping of durable parts, are well suited to make many of the OSHW lab tools mentioned above,[21,22,38] and are commonly used for making enclosures or fixtures for commercially available parts (e.g., a housing for a low-temperature plasma mass spectrometry probe[39]). RepRaps have also been used more creatively with user modifications, such as replacing the plastic extrusion nozzle with other implements such as microscopes[40] or heated syringe pumps (for prototyping paper microfluidics),[41] taking advantage of RepRaps' inexpensive 3D micropositioning abilities. A key advantage in using existing open-source components and tools is that they are easy to obtain (e.g., both Arduino boards and RepRap kits are readily available online or from electronics stores at low cost) and many people are familiar with them.

An important part of the open-source movement is the idea of constantly making and sharing improvements. This does not lend itself well to traditional academic journals, where a publication is normally immutable. Fortunately, a number of services exist to facilitate the development and propagation of open-source hardware and software. One of the most popular sites for hosting and collaborating on open-source hardware and software (currently, millions of projects) is GitHub.[42] GitHub is based on the git version control system (a system designed for tracking changes to a project's files, handling many users at once) and provides free hosting for open-source projects. GitHub allows easy sharing of new developments and the ability to "fork" a project—that is, making a personal copy that can be altered as desired and the changes submitted back to the original project for the original author to consider adopting. Other options include Bitbucket[43] and GitLab,[44] which can be run on a private server, allowing a more personalized site. Another useful instrument for sharing is Open Science Framework,[45] an online tool designed to manage research efforts while making it easy to open projects to the greater scientific community, including both hardware/software designs as well as experimental data, by integrating with external services like GitHub and The Dataverse Project.[46]

Customized Instruments and Software. Perhaps the most enduring aspect of the open-source movement in analytical chemistry is the trend toward developing customized instruments and software that are dedicated to solving particular problems in the laboratory. For example, in the past decade, several OSHW instruments, including electrochemical devices,[26,47−49] gas sensors,[50] and even NMR spectrometer components[51,52] have been described in the literature. During this time, an even greater number of open-source software programs related to analytical chemistry were released. A plethora of open-source software platforms were described in 2016 in *Analytical Chemistry* alone, covering topics from spectral analysis,[53,54] to data fitting,[55] and -omics.[56−63] Also worth noting are the new journals *HardwareX* and *Journal of Open Hardware*, both dedicated to promoting OSHW in all aspects of science, both inside and out of the traditional lab setting;[64,65] presumably, we can expect many publications of interest to analytical chemists in both journals in the coming years. However, these examples are still far outnumbered by articles that describe new hardware or software without releasing them as open source.

While one can find many articles that describe hardware or software that is obviously open- or closed-source, there are also many articles that lie somewhere in between. A common observation in articles written by analytical chemists is a cursory mention of custom software or hardware with no source code and little or no description (cf., refs 66−68). Although these articles describe prototype instruments or software (and perhaps even reference readers to a web page to learn about and use these tools), the connection to end-users is incomplete. We propose that it would be best if authors can learn to release custom developments as FOSS/OSHW, which will allow other researchers to replicate and improve upon the original experiments without designing the setup anew. An interesting example in a recent edition of *Analytical Chemistry* is a web server-based program for handling 2D gas chromatography−mass spectrometry data for metabolomics.[69] The article describes the software and the methods it uses for detecting peaks and correcting baselines, as well as aligning spectra collected under different conditions. This tool is unquestionably useful to a wide audience of analytical chemists, but no source code is available, and the software can only be used by visiting the authors' website. Users can upload data to the service to be processed, but cannot evaluate the source code behind the data processing and there is no guarantee that the service will be available or free of charge indefinitely. This is a good example of material that is usable without payment (popularly phrased as "free, as in beer") but not free to examine, reuse, or modify ("free, as in freedom"). This lack of freedom has obvious disadvantages for the user but also has disadvantages for the creators. The open-source approach permits users to identify and correct bugs or create improvements themselves, submitting them back to the creators, rather than only reporting issues for the original authors to fix (or worse, to simply stop using the service).

Another common "gray" zone (between open- and closed-source) found in the scientific literature is the release of source code or hardware documentation as part of an article or its Supporting Information, but with no explicit license to use the item as FOSS or OSHW (cf., refs 70 and 71). The ability to examine these materials is useful, but a work may not be suitable for direct reuse or improvement by other researchers without the rights granted by a suitable license. Authors always have the option of granting permissions upon request, but this can introduce a significant delay and create uncertainty that may discourage contributions from others. We propose that this may simply be an issue of education, and are hopeful that authors who want to share their tools will find the following discussion to be useful.

CONSIDERATIONS FOR OPEN-SOURCE CREATORS

As noted, articles reporting new hardware or software often either do not provide complete source code and design files or do not release such source files under an explicit license. Without a license, the intentions of the authors may not be clear, leading to ambiguity surrounding permitted usage of the software and hardware designs. Releasing a design under an open-source license provides an explicit legal foundation for authors to maximize use, reuse, propagation, and follow-on contributions—goals that align directly with the sentiment of Newton's vision about the scientific enterprise. Open-source licensing is especially beneficial to scientific research in general where it may facilitate collaboration across institutions, researchers switching institutions, and prolonging the lifetime

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338

Table 1. Summary of Selected Hardware and Software Licenses

license	copyleft[a]	latest version	patent clause[b]	linking allowed[c]	used by
software					
BSD[78]	no	2-clause	no	yes	SciPy,[84] Numpy,[84] IPython[85]
MIT[79]	no	1.0	no	yes	Node.js,[89] Ruby On Rails[90]
Apache[87]	no	2.0	yes	yes	Apache HTTP Server,[91] Android[92]
LGPL[77]	partial[d]	3.0	yes	yes	Arduino,[27] GTK[93]
GPL[76]	yes	3.0	yes	no	Linux kernel,[94] GNOME desktop environment[95]
hardware					
MIT	no	1.0	no		NodeMCU[96]
Creative Commons BY-SA[97]	yes	4.0	no		Arduino,[27] BeagleBoard[98]
GPL	yes	3.0	yes		RepRap,[36] OpenPCR,[99] MultiSpeQ[100]
CERN OHL[101]	yes	1.2	yes		Public Lab Spectrometer,[102] most projects in the Open Hardware Repository[15]

[a]Other projects integrating the original work must be licensed under the same terms as the original work. [b]License terms include explicit rights related to patents. [c]Original work may be integrated into another project licensed under different terms than the original work. [d]When original work is included in another project, only changes to the original work, not the entire project, must be released under the terms of the original work.

of projects beyond the involvement of the original creators.[72] Since the development of instruments for analytical chemistry typically involves both software and hardware components, the licensing considerations of each component must be considered to release the entire project as open source.

Software Licensing. Within an analytical chemistry lab, software may take one of many forms. It may consist of, for example, a simple script used for data processing, a collection of software functions to be reused across applications (i.e., a software library),[73] or a user interface for a custom instrument.[25,26] In the United States and in many other jurisdictions, software is protected under copyright law by default. As such, if an author intends to share software, it is crucial to release it under an explicit license. This is true regardless of software complexity, even applying, perhaps unexpectedly, to "snippets" of code posted in responses on popular question and answer websites such as Stack Overflow.[74,75]

Software licenses are encountered frequently in everyday life. For example, while we may think we are purchasing office-productivity software for our computers or apps for our smartphones, we are in fact not buying the *software* (i.e., source code or executable binary) itself but are rather purchasing the *right* to use the software according to specific license terms—the legalese that most blindly click "OK" to accept. Commercial software is often written in programming languages where the source code must be compiled into a binary form prior to execution. The resulting binary is then distributed to the end user without providing the original source code, such that the user may run the program in its executable form according to the license terms, but without the ability to inspect or modify the software. However, even if the intention is to share source code for others to use, inspect and modify, simply making the code accessible (e.g., publishing it online) does not grant legal rights for others to use the software.

Academic researchers may wish to share software that they have written, perhaps to fulfill open-access publication obligations or just to promote the sharing of knowledge in the spirit of Newton's message. FOSS licenses may be employed to explicitly pass on specific rights to users regarding access and eligible usage. While many different licenses qualify as FOSS, they all share some fundamental properties. First, a FOSS license must apply to all users in a nondiscriminatory fashion, even including use in commercial applications (provided the terms of the license are met). Second, a FOSS license must *at least* explicitly grant users rights to access and inspect the source code, compile (if necessary) and execute the code, modify the code (e.g., to fix bugs, introduce new functionality, incorporate into other projects), and distribute the original source code or any modified version or projects including the licensed code. Licensed source code that is modified or included within another project is referred to as a "derivative work."

FOSS licenses can be categorized into two classes, defined by the conditions under which users may redistribute both the original source code and derivative works. The *first* class is referred to as "copyleft" (as a play on "copyright"), and permits users to freely redistribute the original work and any derivative works provided that any such distributed code is released under *the same licensing terms* (sometimes referred to as "reciprocity"). In other words, if a user would like to modify software that is released under a copyleft license and distribute the result, the modified source code must be provided and released under the same license. The most prominent examples of copyleft licenses include the General Public License (GPL)[76] and Lesser GPL (LGPL),[77] where both require that licensing terms be preserved, but the LGPL relaxes the reciprocity requirement by allowing the original work to be included in another code while only requiring changes to the original work—not the entire project—to be released under the copyleft license. FOSS licenses within the *second* class are referred to as "permissive," in that they grant users the right to release both the original source code and any derivative works under *any licensing terms* that they choose. Permissive licenses allow, for example, users to modify the source code and distribute only the compiled executable, as is common in commercial applications. Examples of permissive licenses include the Berkeley Software Distribution (BSD)[78] and Massachusetts Institute of Technology (MIT)[79] licenses, which stipulate only that adopters include the original copyright notice and a statement of attribution when distributing derivative works.

Given the options (above), how does one choose which license to adopt? Copyleft and permissive licenses target different objectives. While copyleft licenses aim to protect the rights of users of derivative works (including, for example, feeding improvements and bug fixes back into a parent project), permissive licenses aim to maximize the adoption of software by reducing or eliminating license compatibility concerns. A recent survey found that over 58% of open-source software was

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330–4338

licensed under permissive terms, and over 35% licensed under copyleft terms, indicating a significant market share for both classes of FOSS licenses.[80] While permissive licenses allow, for example, derivative works to be released as proprietary commercial applications without requiring the corresponding source code to be released, developers' motivations and beliefs in FOSS principles and redistribution rights may influence license selection more than economic incentive.[81] Moreover, software developers are often motivated to make a selection based on the licensing terms of other projects within their social networks or according to licenses used for similar projects.[82] This phenomenon is evidenced, for example, by the licenses shared by numerous core scientific software packages for the Python[83] programming language, including NumPy,[84] SciPy,[84] IPython,[85] and Matplotlib.[86] Another important consideration when selecting an open-source software license is compatibility with other software applications and libraries. Consider a software library released using a permissive license (e.g., BSD). Such a library may be distributed, for example, as part of another project licensed under a copyleft license (e.g., GPL). However, the reverse is not true; that is, if a developer releases a software library using a copyleft license (e.g., GPL), another project including the library *must* be released under the *same* copyleft license and may not be released under any other license (e.g., BSD license). Therefore, the anticipated usage of software may play a critical role in the selection of a specific FOSS license. In addition to philosophical and compatibility concerns, software patents must also be considered when selecting a license, where applicable. For example, patent grants are explicitly included in the license text of FOSS licenses such as the Apache license.[87]

We hope that the information covered here is a useful introduction to topics discussed herein, but note that it is not a substitute for formal legal advice, and it is important for authors to ensure that any institutional obligations are met. The properties of a number of common software licenses are summarized in Table 1. For further reading, we recommend a review by Morin et al.[72] covering various software licensing options specifically from the perspective of a scientist-programmer, or the choosealicense.com website[88] which aims to simplify license selection.

Hardware Licensing. Licensing of open-source hardware is a complex topic for a number of reasons. A typical open-hardware project is composed of several elements (i.e., digital representations) that are required to replicate a physical object: for example, firmware (i.e., software to run on embedded hardware), schematics, CAD drawings, and documentation. These files may be shared online in much the same way that software developers share source code, but existing licenses and legal concepts developed for open-source software depend on copyright law, and are not directly applicable to hardware.[24,103] For example, while it is possible to copyright a CAD drawing or circuit schematic, legal protection does not extend to the *idea* expressed by that design; instead, the use and manufacture of that design falls under patent law.[103] Although hardware-specific licenses have been developed (e.g., the Tucson Amateur Packet Radio or TAPR License,[104] or the CERN Open Hardware License[101]), many open-hardware projects choose to use more general copyright licenses (e.g., GPL,[76] Creative Commons,[75] or MIT[79]). Copyright-based licenses offer little *real* legal protection to hardware; however, they do serve the important purpose of explicitly signaling the intentions of the original authors/developers to grant permission for others to copy and modify their designs. The properties of a number of common hardware licenses are summarized in Table 1.

Beyond legally enforceable protective measures, informal expectations and social norms also play an important role within the open-source hardware community.[105] Many of these expectations should be familiar to members of the scientific community at large, for example, provide attribution, refrain from plagiarism, demonstrate significant added-value/novelty, etc. A recent certification initiative by the Open Source Hardware Association (OSHWA) seeks to provide a legal framework for enforcing some of these requirements through a certification logo that is protected under trademark law.[106] OSHWA also publishes an open-source hardware definition and statement of principles[13] and a list of "best practices" for open-source hardware projects.[107] This list includes suggestions such as providing original design files that can be used to adapt/modify the design, using free and open-source computer aided design (CAD) tools where possible, and being "emotionally prepared to allow your project to be copied (unless your trademark is violated, then act according to trademark law)."[107] We propose that this list[107] is a good place to start for inventors who are new to the movement and want to participate.

Is Open-Source a Good Fit for My Project? We have primarily focused on the benefits of open source hardware for analytical chemists, but in some cases, the creators of new hardware or software may have good reasons to keep their inventions closed. For example, employees may be legally required to secure permission from their employers before disclosing new inventions since such actions may impact future intellectual property and/or commercialization opportunities. Some creators may also refrain from openly releasing designs to maintain a perceived strategic advantage or they may simply conclude that the extra work involved (e.g., expectations to provide documentation, support, etc.) do not justify the expected benefits. Choosing whether to release a particular invention under an open license is a complex and personal decision that may be strongly influenced by the philosophical beliefs of the creators;[81] however, we believe that as more examples of successful and impactful open projects become apparent, it will increasingly be considered as a viable option by mainstream scientists.

■ DROPBOT/DSTAT

The Wheeler Lab is an analytical chemistry research group (with members that hail from many different disciplines) at the University of Toronto. We have historically enjoyed developing custom hardware and software to solve laboratory problems, and we have recently begun releasing our inventions as open source. Two recent projects produced instruments that were made open source for different reasons: (1) DropBot is a control-system for digital microfluidics, a new instrument in a field where no commercial or other open-source options were available; making DropBot open-source was intended to encourage advancement in the field by providing an alternative to researchers developing their own custom instruments. (2) DStat is a potentiostat, a new instrument in a market saturated with commercial instruments, but the latter are expensive and lack the flexibility for modification that an open-source instrument permits. Each system is described below along with the motivations and benefits behind their design and implementation.

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338

DropBot (and MicroDrop). DropBot is an OSHW digital microfluidic (DMF) automation system designed for general-purpose liquid handling which can be controlled using the MicroDrop FOSS program.[25] The DropBot platform was designed to provide software-controlled movement of droplets on digital microfluidic (DMF) devices, enabling those without electrical engineering and programming backgrounds to conduct DMF research. Prior to the release of DropBot, since there was no off-the-shelf DMF control-system available, researchers were required to build their own systems from scratch. The DropBot is intended as a reference implementation that is accessible to most research laboratories at a modest cost using easy-to-source electrical components, which can optionally be assembled by third-party circuit board manufacturers, significantly lowering the barrier for laboratories that are new to using DMF.

The DropBot system along with the MicroDrop software has been used in many applications including integrated cell culture and analysis,[108] electrochemical immunoassays,[109] and sample preparation for mass spectrometry.[110] Each new application typically requires interfacing with new sensors and other ancillary instruments, so it is crucial that the DropBot platform be extensible. As such, DropBot hardware and MicroDrop software were designed from the outset to promote modularity and to explicitly support interfacing with other hardware through standardized communication interfaces (e.g., interintegrated circuit, or I^2C) and with other software through the introduction of plugins. By releasing both the DropBot hardware and MicroDrop software under open-source licenses, users have the *freedom* and *ability* to make any necessary modifications to the system for a given experimental setup.

The DropBot hardware design files are licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported license,[75] and the DropBot firmware and MicroDrop software are licensed under the BSD[78] software license. The BSD license was chosen to promote adoption of the software by the widest user base possible and to ensure compatibility with the licenses of the core packages in the existing scientific Python[83] ecosystem. As a testament to the value of open source, all DropBot circuit boards were designed using the FOSS KiCad[111] application, the firmware was compiled using the Arduino[27] development environment, and the main interface board was based on an Arduino microcontroller. Moreover, the MicroDrop software was written in Python[83] and relies on numerous open-source packages, including GTK,[93] Numpy,[84] Matplotlib,[86] and many others. It is only by standing on these wide shoulders, raised upon the hard work of others, that the development of the DropBot platform was possible.

The open source nature of the DropBot platform has led to some interesting and exciting opportunities. In 2014, we held the first DropBot DMF workshop, in which attendees from around the world traveled to Toronto to gain hands-on experience constructing and assembling DropBot systems and received training using the MicroDrop software to move droplets first-hand. We also hosted an interactive introductory DMF workshop at the MicroTAS 2016 conference using the latest version of the DropBot platform. Since DropBot and MicroDrop were released as open source in 2013, more than 15 DropBot systems have been built in laboratories around the world and are being used to conduct DMF research. External to our lab, Heinemann et al.[112] recently published work using the DropBot platform, an event that would surely bring a smile to Newton's face.

DStat. DStat is an OSHW general-purpose lab potentiostat with a cross-platform FOSS computer interface.[26] DStat was designed to be accessible to any lab, being inexpensive, compact, and easy-to-use, while maintaining uncompromised measurement performance. The intent of the project was to bridge the gap between very inexpensive potentiostats intended for field use in low-resource settings, like the excellent CheapStat project,[48] and the much more expensive (and closed-source) commercial instruments. A critical motivation in developing DStat was frustration in attempting to interface a commercial potentiostat with DropBot. Commercial instruments generally operate as "black boxes," where little information is available about the hardware and software's internal workings, and options for interfacing other instruments or software varies greatly from instrument to instrument. This was an important consideration in choosing to release DStat as OSHW/FOSS, since making DStat fully open source resolves all of the problems related to the black-box nature of commercial instruments—all design documents and source code is available to the user for examination and modification to suit their needs.

The DStat hardware is licensed under the CERN Open Hardware License[101] and the microcontroller firmware and computer interface are licensed under the GNU GPL.[76] These licenses were chosen to encourage users to contribute modifications back to the community, as both licenses stipulate that modifications must be released under the original license. This has the added effect of discouraging the use of the work in commercial products without acquiring a separate license from the author, as most companies prefer not to release intimate details about their products' designs (leading to the aforementioned black-box problem). In addition, DStat is built using existing open-source tools: the circuit board designs were produced with KiCad,[111] the microcontroller firmware was written in C using the avr-libc library,[113] and much like Microdrop, the computer interface is written in Python[83] with a number of open-source modules like GTK[93] and NumPy.[84] Following the theme of this article, DStat would not have been possible without standing on the shoulders of these FOSS tools and frameworks, which are the result of years of development shared by hundreds of people.

Releasing DStat as OSHW/FOSS has allowed others to rapidly replicate our efforts—since its release in late 2015, several individuals, academic groups, and companies have expressed interest in DStat, a number of whom have constructed their own DStats and submitted modifications back to the project. Just as we built upon open-source projects to produce DStat, we wish to encourage others to build upon DStat to improve the design and modify it to suit their own needs. While a traditional journal publication (without providing the design files, source code, and licenses granting permissions to use and modify) might have inspired some of the above, in all likelihood, the instrument would have remained within our lab.

THE FUTURE OF OPEN SOURCE IN ANALYTICAL CHEMISTRY

Open-source hardware stands to benefit many stakeholders within the scientific ecosystem, for example, scientists as users, scientists as developers, funding agencies, and the public at large. Oft-cited benefits include reduced cost,[24] better reproducibility,[114] and democratization of technology (i.e., making it easier for those outside of traditional institutions to

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338

access the necessary tools to practice science). For analytical chemists, founding or contributing to an open-source hardware project is likely to bring increased publicity and exposure to one's work (analogous to the well-documented open-access citation advantage[115]) and can bring valuable feedback (with the potential to attract new contributions). It is also easy to imagine many indirect benefits that can result from having more open hardware in the analytical chemistry lab; for example, giving researchers the capability to better understand their own tools may provide the means to tackle new and interesting questions that would otherwise be inaccessible using proprietary "black box" systems.

If we accept the premise that broader adoption of open-source hardware (both development and use) will benefit scientists and society in general, the following questions remain: what tools exist for increasing adoption and what lessons can be learned from previous *open* movements? The community-developed GOSH manifesto provides a statement of principles for open-source hardware in science, including such topics such as accessibility, empowerment, scientific impact, democratization, and changing the culture of science.[14] This last point, changing cultural norms, is particularly important if open-source hardware is to be widely adopted within the field of analytical chemistry and in academia in general. Traditional academic incentives encourage the production of "novel" findings over the development of tools and improvements to the overall efficiency of the scientific enterprise. Although some scientists may be intrinsically motivated to release their work under open licenses, many more are likely to do so if the system provides the right extrinsic incentives; therefore, policy incentives are likely to be an important tool for encouraging widespread development of open-source hardware and software in analytical chemistry and beyond. These incentives may be implemented at various administrative levels (e.g., academic journals, national governments, funding bodies, and research institutions) and we need only look to other, more established *open* movements (e.g., open-source software, open publishing, and open data) for inspiration.

Many granting agencies (e.g., NIH,[116] and Wellcome Trust[117]) require that articles associated with any of their grants be published in open-access journals immediately (or deposited in open repositories after a short embargo period). Mandates for open-access publishing have also been implemented at the institutional level.[115] Another interesting example of creative policy incentives is the "Open Practices" badge system, recently proposed by the Center for Open Science.[118] Through this policy, publishers are encouraged to apply a graphical badge to articles containing open data, open materials, etc. By adopting such a system, journals can signal their support of these practices. There is already some evidence that such badges can have a positive effect; in a study of one journal (*Psychological Science*), open-data badges were shown to increase data-sharing more than 10-fold.[118] In theory, such a system could be easily extended to cover open-source software and hardware. Implementing broad requirements that force researchers to use open-source hardware seems unreasonable at this stage; however, it may be feasible to create special grants that require researchers to release their hardware designs under an accepted open-source license. These grants could even be strategically targeted to the production of open-source instruments that would be of long-term benefit to a particular agency or institution. In any case, the availability of a toolbox of high-quality, open-source instruments is likely to have many positive downstream effects (e.g., reducing purchasing costs and enhancing the productivity of researchers making hardware improvements), thus representing a strong economic and scientific return on investment.

Finally, the purported benefits of open-source hardware on analytical chemistry (and science in general) can only be realized if open-source instruments and tools can be produced and supported in ways that are competitive with conventional instrument suppliers, both economically and in terms of analytical performance and reproducibility. While there is already an established commercial ecosystem around open-source software, the same cannot yet be said for the nascent open-source hardware community. The development of successful business models and services (e.g., calibration, support, customization) around open-source scientific hardware will be critical to its long-term impact.

Science has changed since the time of Newton, but the principle of building upon the works of others has not. As we move toward ever increasing complexity in our experiments, it is incumbent upon us to consider ideas like the open-source concept to preserve this vision for the generations of scientists to come.

AUTHOR INFORMATION

Corresponding Author

*E-mail: aaron.wheeler@utoronto.ca. Phone: +1(416) 946-3864.

ORCID

Aaron R. Wheeler: 0000-0001-5230-7475

Notes

The authors declare no competing financial interest.

REFERENCES

(1) arXiv Monthly Submission Rates. https://arxiv.org/stats/monthly_submissions (accessed 2016/10/04).

(2) bioRxiv.org—the preprint server for Biology. http://biorxiv.org/ (accessed 2016/11/17).

(3) American Chemical Society announces intention to establish "ChemRxiv" preprint server to promote early research sharing. https://www.acs.org/content/acs/en/pressroom/newsreleases/2016/august/acs-announces-intention-to-establish-chemrxiv-preprint-server-to-promote-early-research-sharing.html (accessed 2016/11/17).

(4) ACS Omega: About the Journal. http://pubs.acs.org/page/acsodf/about.html (accessed 2016/10/04).

(5) Netcraft September 2016 Web Server Survey. https://news.netcraft.com/archives/2016/09/19/september-2016-web-server-survey.html (accessed 2016/10/06).

(6) Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchison, G. R. *J. Cheminf.* **2012**, *4*, 17.

(7) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. *J. Cheminf.* **2011**, *3*, 33.

(8) Turney, J. M.; Simmonett, A. C.; Parrish, R. M.; Hohenstein, E. G.; Evangelista, F. A.; Fermann, J. T.; Mintz, B. J.; Burns, L. A.; Wilke, J. J.; Abrams, M. L.; Russ, N. J.; Leininger, M. L.; Janssen, C. L.; Seidl, E. T.; Allen, W. D.; Schaefer, H. F.; King, R. A.; Valeev, E. F.; Sherrill, C. D.; Crawford, T. D. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2012**, *2*, 556−565.

(9) Valiev, M.; Bylaska, E. J.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Van Dam, H. J. J.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T. L.; de Jong, W. A. *Comput. Phys. Commun.* **2010**, *181*, 1477−1489.

(10) Janssen, C. L.; Nielsen, I. B.; Leininger, M. L.; Valeev, E. F.; Kenny, J. P.; Seidl, E. T. The Massively Parallel Quantum Chemistry Program (MPQC), 2008. http://www.mpqc.org/index.php (accessed 2017/01/20).

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338

(11) Uppal, K.; Walker, D. I.; Jones, D. P. *Anal. Chem.* **2017**, *89*, 1063−1067.
(12) Kirkpatrick, C. L.; Broberg, C. A.; McCool, E. N.; Lee, W. J.; Chao, A.; McConnell, E. W.; Pritchard, D. A.; Hebert, M.; Fleeman, R.; Adams, J.; Jamil, A.; Madera, L.; Strömstedt, A. A.; Göransson, U.; Liu, Y.; Hoskin, D. W.; Shaw, L. N.; Hicks, L. M. *Anal. Chem.* **2017**, *89*, 1194−1201.
(13) Open Source Hardware Definition (English). http://www.oshwa.org/definition/ (accessed 2016/10/06).
(14) GOSH Manifesto. http://openhardware.science/gosh-manifesto/ (accessed 2016/10/06).
(15) Open Hardware Repository. http://www.ohwr.org/projects (accessed 2016/10/06).
(16) Baden, T.; Chagas, A. M.; Gage, G. J.; Marzullo, T. C.; Prieto-Godino, L. L.; Euler, T. *PLoS Biol.* **2015**, *13*, e1002086.
(17) NIH 3D Print Exchange. https://3dprint.nih.gov (accessed 2017/02/28).
(18) Open Source Toolkit: Hardware. http://collections.plos.org/open-source-toolkit-hardware (accessed 2017/02/28).
(19) Wittbrodt, B. T.; Squires, D. A.; Walbeck, J.; Campbell, E.; Campbell, W. H.; Pearce, J. M. *PLoS One* **2015**, *10*, e0134989.
(20) Mista, C.; Zalazar, M.; Peñalva, A.; Martina, M.; Reta, J. M. *J. Phys.: Conf. Ser.* **2016**, *705*, 012008.
(21) Open-Source Test Tube Racks. http://www.appropedia.org/Category:Open_Source_test_tube_racks (accessed 2016/10/07).
(22) Open-Source Centrifuges. http://www.appropedia.org/Category:Open_source_centrifuges (accessed 2016/10/07).
(23) Brennan, M.; Bokhari, F.; Eddington, D. Open Design 3D-Printable Adjustable Micropipette that meets ISO Standard for Accuracy. 2017, 109231. bioRxiv.org Preprint Server for Biology. http://biorxiv.org/content/early/2017/02/20/109231.article-info.
(24) Pearce, J. M. *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*; Elsevier, 2013.
(25) Fobel, R.; Fobel, C.; Wheeler, A. R. *Appl. Phys. Lett.* **2013**, *102*, 193513−193518.
(26) Dryden, M. D. M.; Wheeler, A. R. *PLoS One* **2015**, *10*, e0140349.
(27) Arduino. Introduction. https://www.arduino.cc/en/Guide/Introduction (accessed 2016/12/05).
(28) Bürgel, S. C.; Diener, L.; Frey, O.; Kim, J.-Y.; Hierlemann, A. *Anal. Chem.* **2016**, *88*, 10876−10883.
(29) Velusamy, V.; Arshak, K.; Korostynska, O.; Al-Shamma'a, A. *Key Eng. Mater.* **2013**, *543*, 47−50.
(30) Miranda, J. C.; Kamogawa, M. Y.; Reis, B. F. *Sens. Actuators, B* **2015**, *207*, 811−818.
(31) Chiu, S. H.; Urban, P. L. *Biosens. Bioelectron.* **2015**, *64*, 260−268.
(32) Mai, T. D.; Le, M. D.; Sáiz, J.; Duong, H. A.; Koenka, I. J.; Pham, H. V.; Hauser, P. C. *Anal. Chim. Acta* **2016**, *911*, 121−128.
(33) Grinias, J. P.; Whitfield, J. T.; Guetschow, E. D.; Kennedy, R. T. *J. Chem. Educ.* **2016**, *93*, 1316−1319.
(34) Shih, S. C. C.; Yang, H.; Jebrail, M. J.; Fobel, R.; McIntosh, N.; Al-Dirbashi, O. Y.; Chakraborty, P.; Wheeler, A. R. *Anal. Chem.* **2012**, *84*, 3731−3738.
(35) da Costa, E. T.; Mora, M. F.; Willis, P. A.; do Lago, C. L.; Jiao, H.; Garcia, C. D. *Electrophoresis* **2014**, *35*, 2370−2377.
(36) About RepRapWiki. http://reprap.org/wiki/About (accessed 2016/12/08).
(37) Duarte, L. C.; Colletes de Carvalho, T.; Lobo-Júnior, E. O.; Abdelnur, P. V.; Vaz, B. G.; Coltro, W. K. T. *Anal. Methods* **2016**, *8*, 496−503.
(38) Biological Microsystems Laboratory, Github. https://github.com/Biological-Microsystems-Laboratory (accessed 2017/01/02).
(39) Martínez-Jarquín, S.; Moreno-Pedraza, A.; Guillén-Alonso, H.; Winkler, R. *Anal. Chem.* **2016**, *88*, 6976−6980.
(40) Wijnen, B.; Petersen, E. E.; Hunt, E. J.; Pearce, J. M. *J. Microsc. (Oxford, U. K.)* **2016**, *264*, 238−246.
(41) Pearce, J. M.; Anzalone, N. C.; Heldt, C. L. *J. Lab. Autom.* **2016**, *21*, 510−516.
(42) Github: Where open source lives. https://github.com/open-source (accessed 2017/01/23).
(43) Bitbucket. https://bitbucket.org (accessed 2017/01/26).
(44) GitLab. https://about.gitlab.com (accessed 2017/01/26).
(45) Open Science Framework. https://osf.io (accessed 2017/03/10).
(46) Dataverse Project. http://dataverse.org (accessed 2017/03/10).
(47) Friedman, E. S.; Rosenbaum, M. A.; Lee, A. W.; Lipson, D. A.; Land, B. R.; Angenent, L. T. *Biosens. Bioelectron.* **2012**, *32*, 309−313.
(48) Rowe, A. A.; Bonham, A. J.; White, R. J.; Zimmer, M. P.; Yadgar, R. J.; Hobza, T. M.; Honea, J. W.; Ben-Yaacov, I.; plaxco, K. W. *PLoS One* **2011**, *6*, e23783.
(49) Salvador, C.; Mesa, M. S.; Durán, E.; Alvarez, J. L.; Carbajo, J.; Mozo, J. D. *Rev. Sci. Instrum.* **2016**, *87*, 055111.
(50) Gotor, R.; Gaviña, P.; Costero, A. M. *Meas. Sci. Technol.* **2015**, *26*, 085103.
(51) Takeda, K. *J. Magn. Reson.* **2008**, *192*, 218−229.
(52) Coffey, A. M.; Shchepin, R. V.; Truong, M. L.; Wilkens, K.; Pham, W.; Chekmenev, E. Y. *Anal. Chem.* **2016**, *88*, 8279−8288.
(53) Race, A. M.; Palmer, A. D.; Dexter, A.; Steven, R. T.; Styles, I. B.; Bunch, J. *Anal. Chem.* **2016**, *88*, 9451−9458.
(54) Mahieu, N. G.; Spalding, J. L.; Gelman, S. J.; Patti, G. J. *Anal. Chem.* **2016**, *88*, 9037−9046.
(55) Fourmond, V. *Anal. Chem.* **2016**, *88*, 5050−5052.
(56) Liang, Y. J.; Lin, Y. T.; Chen, C. W.; Lin, C. W.; Chao, K. M.; Pan, W. H.; Yang, H. C. *Anal. Chem.* **2016**, *88*, 6334−6341.
(57) Capellades, J.; Navarro, M.; Samino, S.; Garcia-Ramirez, M.; Hernandez, C.; Simo, R.; Vinaixa, M.; Yanes, O. *Anal. Chem.* **2016**, *88*, 621−628.
(58) Sun, R. X.; Luo, L.; Wu, L.; Wang, R. M.; Zeng, W. F.; Chi, H.; Liu, C.; He, S. M. *Anal. Chem.* **2016**, *88*, 3082−3090.
(59) Domingo-Almenara, X.; Brezmes, J.; Vinaixa, M.; Samino, S.; Ramirez, N.; Ramon-Krauel, M.; Lerin, C.; Díaz, M.; Ibáñez, L.; Correig, X.; Perera-Lluna, A.; Yanes, O. *Anal. Chem.* **2016**, *88*, 9821−9829.
(60) Kochen, M. A.; Chambers, M. C.; Holman, J. D.; Nesvizhskii, A. I.; Weintraub, S. T.; Belisle, J. T.; Islam, M. N.; Griss, J.; Tabb, D. L. *Anal. Chem.* **2016**, *88*, 5733−5741.
(61) Collins, J. R.; Edwards, B. R.; Fredricks, H. F.; Van Mooy, B. A. S. *Anal. Chem.* **2016**, *88*, 7154−7162.
(62) Yılmaz, Ş.; Drepper, F.; Hulstaert, N.; Černič, M.; Gevaert, K.; Economou, A.; Warscheid, B.; Martens, L.; Vandermarliere, E. *Anal. Chem.* **2016**, *88*, 9949−9957.
(63) Yu, C. Y.; Mayampurath, A.; Zhu, R.; Zacharias, L.; Song, E.; Wang, L.; Mechref, Y.; Tang, H. *Anal. Chem.* **2016**, *88*, 5725−5732.
(64) HardwareX, Journal, Elsevier. https://www.journals.elsevier.com/hardwarex/ (accessed 2017/01/02).
(65) Journal of Open Hardware. http://openhardware.metajnl.com (accessed 2017/03/24).
(66) Vaillier, C.; Honegger, T.; Kermarrec, F.; Gidrol, X.; Peyrade, D. *Anal. Chem.* **2016**, *88*, 9022−9028.
(67) Lewenstam, A.; Blaz, T.; Migdalski, J. *Anal. Chem.* **2017**, *89*, 1068−1072.
(68) Conzuelo, F.; Sliozberg, K.; Gutkowski, R.; Grützke, S.; Nebel, M.; Schuhmann, W. *Anal. Chem.* **2017**, *89*, 1222−1228.
(69) Tian, T.-F.; Wang, S.-Y.; Kuo, T.-C.; Tan, C.-E.; Chen, G.-Y.; Kuo, C.-H.; Chen, C.-H. S.; Chan, C.-C.; Lin, O. A.; Tseng, Y. J. *Anal. Chem.* **2016**, *88*, 10395−10403.
(70) Misawa, T.; Wei, F.; Kikuchi, J. *Anal. Chem.* **2016**, *88*, 6130−6134.
(71) Guder, J. C.; Schramm, T.; Sander, T.; Link, H. *Anal. Chem.* **2017**, *89*, 1624−1631.
(72) Morin, A.; Urban, J.; Sliz, P. *PLoS Comput. Biol.* **2012**, *8*, e1002598.
(73) Safta, C.; Najm, H. N.; Knio, O. TChem. http://www.sandia.gov/tchem/ (accessed 2017/01/26).
(74) Attribution Required. https://stackoverflow.blog/2009/06/attribution-required/ (accessed 2017/01/26).

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338

(75) Attribution-ShareAlike 3.0 Unported. https://creativecommons.org/licenses/by-sa/3.0/legalcode (accessed 2017/01/26).
(76) The GNU General Public License. https://www.gnu.org/licenses/gpl-3.0.en.html (accessed 2017/01/08).
(77) The GNU Lesser General Public License. https://www.gnu.org/licenses/lgpl-3.0.en.html (accessed 2017/01/12).
(78) The 3-Clause BSD License|Open Source Initiative. https://opensource.org/licenses/BSD-3-Clause (accessed 2017/01/20).
(79) The MIT License|Open Source Initiative. https://opensource.org/licenses/MIT (accessed 2017/01/20).
(80) Top Open Source Licenses. https://www.blackducksoftware.com/top-open-source-licenses (accessed 2017/01/26).
(81) Sen, R.; Subramaniam, C.; Nelson, M. L. *J. Manage. Inform. Syst.* **2008**, *25*, 207−240.
(82) Singh, P. V.; Phelps, C. *Inf. Syst. Res.* **2013**, *24*, 539−560.
(83) Welcome to Python.org. https://www.python.org/ (accessed 2017/01/12).
(84) van der Walt, S.; Colbert, S. C.; Varoquaux, G. *Comput. Sci. Eng.* **2011**, *13*, 22−30.
(85) Perez, F.; Granger, B. E. *Comput. Sci. Eng.* **2007**, *9*, 21−29.
(86) Hunter, J. D. *Comput. Sci. Eng.* **2007**, *9*, 90−95.
(87) Apache License, Version 2.0. http://www.apache.org/licenses/LICENSE-2.0.html (accessed 2017/01/12).
(88) Choose an open source license. https://choosealicense.com/ (accessed 2017/01/26).
(89) Node.js. https://nodejs.org (accessed 2017/03/10).
(90) Ruby on Rails. http://rubyonrails.org (accessed 2017/03/10).
(91) The Apache HTTP Server Project. https://httpd.apache.org (accessed 2017/03/10).
(92) Android Open Source Project. https://source.android.com (accessed 2017/03/10).
(93) The GTK= Project. https://www.gtk.org/ (accessed 2017/01/12).
(94) The Linux Kernel Archives. https://www.kernel.org (accessed 2017/03/10).
(95) GNOME. https://www.gnome.org/ (accessed 2017/03/10).
(96) NodeMCU. https://github.com/nodemcu/nodemcu-devkit-v1.0 (accessed 2017/03/10).
(97) Creative Commons—CC BY-SA 3.0. https://creativecommons.org/licenses/by-sa/3.0/ (accessed 2017/02/28).
(98) BeagleBoard. https://beagleboard.org/ (accessed 2017/03/10).
(99) OpenPCR. http://openpcr.org (accessed 2017/03/10).
(100) Kuhlgert, S.; Austic, G.; Zegarac, R.; Osei-Bonsu, I.; Hoh, D.; Chilvers, M. I.; Roth, M. G.; Bi, K.; TerAvest, D.; Weebadde, P.; Kramer, D. M. R. *R. Soc. Open Sci.* **2016**, *3*, 160592.
(101) CERN Open Hardware License. http://www.ohwr.org/projects/cernohl/wiki (accessed 2017/01/08).
(102) Public Lab Spectrometer. https://publiclab.org/wiki/spectrometer (accessed 2017/03/10).
(103) Open Source Hardware and the Law. https://www.publicknowledge.org/news-blog/blogs/open-source-hardware-and-law (accessed 2017/01/12).
(104) The TAPR Open Hardware License. http://www.tapr.org/ohl.html (accessed 2017/01/12).
(105) Soapbox: The Unspoken Rules of Open Source Hardware. http://makezine.com/2012/02/14/soapbox-the-unspoken-rules-of-open-source-hardware/ (accessed 2017/01/12).
(106) Open Source Hardware Certificate. http://certificate.oshwa.org/ (accessed 2017/01/12).
(107) Best Practices for Open-Source Hardware 1.0. http://www.oshwa.org/sharing-best-practices/ (accessed 2017/01/12).
(108) Ng, A. H.; Li, B. B.; Chamberlain, M. D.; Wheeler, A. R. *Annu. Rev. Biomed. Eng.* **2015**, *17*, 91−112.
(109) Rackus, D. G.; Shamsi, M. H.; Wheeler, A. R. *Chem. Soc. Rev.* **2015**, *44*, 5320−5340.
(110) Kirby, A. E.; Wheeler, A. R. *Anal. Chem.* **2013**, *85*, 6178−6184.
(111) KiCad EDA. http://kicad-pcb.org/ (accessed 2017/01/12).
(112) Heinemann, J.; Deng, K.; Shih, S. C. C.; Gao, J.; Adams, P. D.; Singh, A. K.; Northen, T. R. *Lab Chip* **2017**, *17*, 323−331.
(113) AVR Libc Home Page. http://www.nongnu.org/avr-libc/ (accessed 2017/01/13).
(114) Munafó, M. R.; Nosek, B. A.; Bishop, D. V. M.; Button, K. S.; Chambers, C. D.; Percie du Sert, N.; Simonsohn, U.; Wagenmakers, E.-J.; Ware, J. J.; Ioannidis, J. P. A. *Nat. Hum. Behav.* **2017**, *1*, 0021.
(115) Gargouri, Y.; Hajjem, C.; Lariviére, V.; Gingras, Y.; Carr, L.; Brody, T.; Harnad, S. *PLoS One* **2010**, *5*, e13636.
(116) When and How to Comply. https://publicaccess.nih.gov/ (accessed 2017/01/12).
(117) Open access policy. https://wellcome.ac.uk/funding/managing-grant/open-access-policy (accessed 2017/01/12).
(118) Kidwell, M. C.; Lazarević, L. B.; Baranski, E.; Hardwicke, T. E.; Piechowski, S.; Falkenberg, L.-S.; Kennett, C.; Slowik, A.; Sonnleitner, C.; Hess-Holden, C.; Errington, T. M.; Fiedler, S.; Nosek, B. A. *PLoS Biol.* **2016**, *14*, e1002456.

DOI: 10.1021/acs.analchem.7b00485
Anal. Chem. 2017, 89, 4330−4338