Technical Note

Open-source Assisted Laboratory Automation through Graphical User Interfaces and 3D Printers. Application to Equipment Hyphenation for Higher-order data Generation

Gabriel Germán Siano, Milagros Montemurro, Mirta Raquel Alcaráz, and Hector Casimiro Goicoechea

Anal. Chem., **Just Accepted Manuscript** • DOI: 10.1021/acs.analchem.7b02758 • Publication Date (Web): 13 Sep 2017

Open-source Assisted Laboratory Automation through Graphical User Interfaces and 3D Printers. Application to Equipment Hyphenation for Higher-order data Generation

Gabriel G. Siano,[a,b,1] Milagros Montemurro,[a,b] Mirta R. Alcaráz,[a,b] and Héctor C. Goicoechea[a,b,*]

[a] *Laboratorio de Desarrollo Analítico y Quimiometría, Cátedra de Química Analítica I, Facultad de Bioquímica y Ciencias Biológicas, Universidad Nacional del Litoral, Ciudad Universitaria, S3000ZAA Santa Fe, Argentina*

[b] *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Godoy Cruz 2290 CABA (C1425FQB), Argentina.*

[1] Corresponding authors: gabrielsiano@gmail.com (G.G. Siano) and hgoico@fbcb.unl.edu.ar (H.C. Goicoechea). Tel: +54 342 4575206x190

ORCID
Gabriel Siano: 0000-0002-6485-4211
Héctor Goicoechea: 0000-0001-7145-0082

ABSTRACT

Higher-order data generation implies some automation challenges, which are mainly related to the hidden programming languages and electronic details of the equipment. When techniques and/or equipment hyphenation is the key to obtain higher-order data, the required simultaneous control of them demands funds for new hardware, software and licenses, and also really skilled operators.

In this work we present Design of Inputs-Outputs with Sikuli (D.I.O.S.), a free and open-source code program that provides a general framework for the design of automated experimental procedures without prior knowledge of programming or electronics. Basically, instruments and devices are considered as nodes in a network, and every node is associated both with physical and virtual inputs and outputs. Virtual components, such as Graphical User Interfaces of equipment, are handled by means of image recognition tools provided by Sikuli scripting language, while handling of their physical counterparts is achieved using an adapted open-source 3D Printer.

Two previously reported experiments of our research group, related to fluorescence matrices derived from kinetics and high performance liquid chromatography, were adapted to be carried out in a more automated fashion. Satisfactory results, in terms of analytical performance, were obtained. Likewise, advantages derived from open-source tools assistance could be appreciated, mainly in terms of lesser intervention of operators and of funds savings.

KEYWORDS

Open-source, higher-order data, chemometrics, Sikuli, Arduino, Rep-Rap

In analytical chemistry, the modeling of higher-order data and its inclusion in multiway calibration models is a field worth exploring. As examples, it has been already provided fourth and fifth-way calibration solutions for quantitation of pesticides[1] and herbicides,[2] respectively. The fact that the concentration of calibrated analytes can be estimated even in the presence of uncalibrated substances is a revolutionary advantage that belongs to second-order calibration (that is the so-called second-order advantage),[3] and it is also present in higher-order calibration models. Hence, the potential of multiway calibration mainly relies in this advantage, although further advantages could be found. For instance, from the fact that the combination of signals from multiple orders implies an increase in the information leading to an improvement in the figures of merit, such as selectivity and sensitivity, among others.[4] Recently, an example involving complex samples in which four-way calibration models clearly achieved better results than lower-order models was reported.[5] Based on those achievements, the authors suggested the existence of a third-order advantage, especially, the selectivity enhancement.

Frequently, the generation of this kind of data involves repeated actions or combination of instruments, which require really skilled operators or a high level of automation. Assuming proper funds are available, automation can be solved with proprietary accessories and licenses for instruments. If this is not the case, it will involve significant challenges.

The knowledge of the languages in which instruments are programmed could be needed to execute functions or to coordinate them with others instruments, as well as their electronics details. Nevertheless, that knowledge will be useful only for those instruments, non-transferable to other ones. Beyond the fact that some kind of hesitation arises when adaptation or modification of instruments are needed, such interventions are highly inadvisable by commercial technical support, or even prohibited when guarantees are in force. Finally, even having succeeded, the experimental results can only be reproduced using exactly the same instrumental configuration, which represents a major drawback in terms of versatility. That is, a strong dependence on specific technology and a loss of generalizability when designing experiments.

Considering that nowadays most of the instruments are handled by computer programs, the most basic way to interact with them relies on their Graphical User Interfaces (GUI). Therefore, applying image recognition

techniques on them, by means of tools such as Sikuli[6] scripting language, represents an alternative way to control interactions with equipment. This kind of programming is extremely generic, with a high level of abstraction, and, since it is mostly visual and intuitive, it is easy to perform even by inexperienced users.

The open-source paradigm is on the opposite side of the black-box model adopted by classical technologies. Free and open source (or *libre*) technological development is a fundamentally new, decentralized, participatory and transparent system to create both software and hardware.[7] Free and Open-Source Software (FOSS) is a computer software made available as source code (open source) that can be used, studied, copied, modified, and redistributed without restriction, or with restrictions that only ensure that further recipients have the same rights under which it was obtained.[8] Advantages of FOSS include lesser dependence on proprietary systems, efficiency of global collaboration, demand-driven innovation, among others. In hardware terms, while a design can be copied, it is not possible to copy a piece of hardware as freely as if it were software. Beyond that practical difference, the general FOSS concept can be expanded to hardware (FOSH). As stated in Pearce's book,[8] high-cost scientific equipment can be fabricated using digital designs at factor of 10–100 cost reductions. Perspectives of these and other related concepts, such as kind of licenses, were recently reported in specific relation to analytical chemistry.[9] Also, in the last two citations there are detailed descriptions and application cases of two successful examples of FOSH that were used in our work: Arduino[10] platforms and RepRap[11] 3D Printers.

In this work, we present third-order data generation examples, based on previous experiments carried out in our laboratory, but performing them with a higher level of automation. We introduce Design of Inputs-Outputs with Sikuli (D.I.O.S.), a program that provides a new general framework to design automated experimental procedures, in which instruments and devices are conceived as network units with Inputs and Outputs (IO), having both physical ones, such as emission readings or drops discharged by a HPLC, and virtual ones, such as mouse and keyboard actions, display changes, etc. Automation of GUIs is performed through embedding Sikuli scripts, whereas the interaction with physical IO is accomplished by means of the 3D positioning features of a 3D Printer handled by D.I.O.S. and assisted by 3D printed pieces complemented with electronic, metallic and

magnetic components. The resulting setup allows the users to hyphenate/coordinate equipment without requiring prior knowledge of programming languages or electronic details.

EXPERIMENTAL SECTION

Two analytical methodologies, previously reported by our research group, were adapted to be done in a more automated fashion. One consisted of a HPLC-EEM system,[12] in which third-order data were generated by fraction collection at the end of the HPLC procedure with posterior EEM registering of each fraction in a plate reader. Here, the operator was responsible of synchronizing injection and collection starting, and of transporting plates from the collection platform to the plate reader. The second case refers to a Kinetics-EEM system,[13] in which EEM data were obtained by photo-induced degradation of an analyte in alkaline medium, collecting several EEM at different times of irradiation. The operator was in charge of the NaOH addition and also of starting and stopping both irradiation and registering cycles. Details on adaptations for each experiment will be covered in the text (also in Supporting Information), besides the reader is encouraged to read those reports. Materials and reagents were the same as reported, as well as HPLC and irradiation, in terms of instruments, programs and procedures. Details and different applications of EEMs can be seen in Reference 4.

To automate experiences, several devices were developed in our laboratory. Most of them were made with 3D-printed parts and open-source electronic platforms. Details on assembly and usage instructions, 3D design files and programs code is available at www.fbcb.unl.edu.ar/laboratorios/ladaq/open-source.

Apparatus

3D Printer

A Prusa i3[14] model was built in our lab, using Marlin[15] as 3D Printer firmware. A 3D-printed piece, hereafter named “the hand”, was designed and coupled to the 3D Printer carriage. Some neodymium magnets were inserted in the hand with the purpose of coupling with metal components, which were present in the bottom part of holders for tools that were moved during experiments. The hand, holders and other 3D-printed parts can be seen in Figure S-1.

HPLC-EEM system adaptations

EEM were recorded using a bifurcated optical fiber, which was optically mounted to a fluorimeter through a lab-made 3D-printed holder for cuvettes and optical fibers. Collection of fractions and fluorescence readings were automated in a 96-well plate, which was over the mobile printing platform of the 3D Printer. The fraction collection started 35 seconds after detecting sample injection (100 μL) and the capillary stayed 1.5 seconds in each well (12 per sample). The arrangement of the HPLC, the 3D Printer and the fluorimeter is depicted in Scheme 1. More details can be obtained from Supporting Information.

Kinetics-EEM system adaptations

Addition of NaOH was automated by using a lab-made drop counter, handled by an Arduino UNO board and a GUI written in Python 2.7.[16] In order to magnetically stir samples, the aforementioned 3D-printed holder for cuvettes also contains stirring capabilities. Different speeds were set through code sent from the Arduino Serial Monitor. More details can be obtained from Supporting Information.

Software

General programs

PARAFAC[17] modeling was implemented through MVC3[18] running in MATLAB.[19] Sikuli scripts were developed with SikuliX IDE,[20] Python code with PyCharm,[21] and Arduino code with Arduino IDE. 3D designs were developed with SketchUp,[22] sliced with Cura[23] and printed with Pronterface.[24]

Design of Inputs-Outputs with Sikuli

D.I.O.S. is a program developed in our laboratory, which source code has been released to be free used under the terms of the AGPL-3.[25] It is written in Python 2.7 and it has Server-Clients structure. Control of GUIs, based on image recognition techniques, is implemented with SikuliX. Tasks over defined GUIs are graphically automated using SikuliX IDE and used later by the Server.

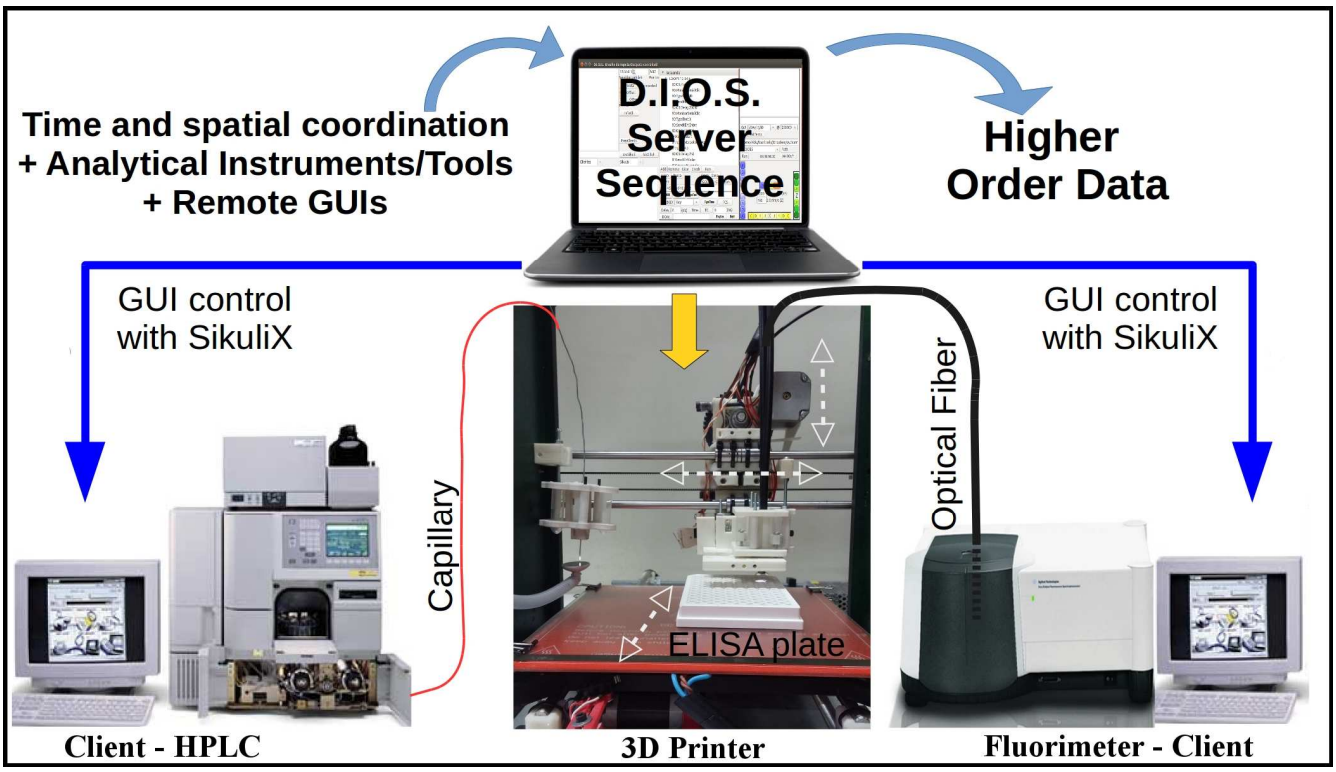
D.I.O.S. Server features can be divided into three categories, each one represented by a panel in its GUI:

1) Communication with Clients: through LAN

2) Communication with 3D Printer: D.I.O.S. Server works with the 3D printing software Pronsole.[24] Any Computer Numerical Control (CNC) equipment able to be controlled through G-code scripting language can be handled by the Server.

3) D.I.O.S. Sequence: In this panel, one can design a Sequence of tasks that have to be made by one or more Clients (remote execution of SikuliX), the 3D Printer (G-codes execution) or the D.I.O.S. Server

Details on features, installation instructions and the software code can be obtained from www.fbcb.unl.edu.ar/laboratorios/ladaq/open-source. Images of D.I.O.S. Server and Client GUIs can be seen in Figures S-2 and S-3, respectively. A schematic diagram of D.I.O.S. features can be seen in Scheme 1.



Scheme 1. D.I.O.S. features

D.I.O.S. Sequences

The HPLC-EEM sequence can be seen in Scheme 2, written in pseudo-code for understanding purposes.

```
Loop samples 1:8
  G-code: Take CT and wait above drain funnel
  Sikuli: HPLC GUI: set number of sample (vial);
  request injection; wait until GUI confirms injection
  Start Time Coordination (35")
  G-code: move CT, align it with respective plate column
  Finish Time Coordination (35"): Wait if needed
  G-code: move CT to the first well, stay 1.5" and repeat
  for wells in the same column (11); hang CT; Take OFT
  Loop well 1:12
    G-code: move OFT above a well
    Sikuli: Fluorimeter GUI: request new matrix scan
  G-code: hang OFT
Server,Printer,Client,CT:Capillary Tool,OFT:Optical Fiber Tool
```

Scheme 2**.** D.I.O.S. Sequence for HPLC-EEM system (pseudo-code)

Looking at the HPLC-EEM sequence, the first line for each sample involves taking the capillary and waiting in a safe place, since it is always discharging drops of HPLC mobile phase (no flow interruption is carried out during the whole experiment). Then, injection is requested by mouse and keyboard actions in the HPLC GUI, followed by image analysis to detect the exact moment at which injection occurs. Immediately after that, a Time Coordination (TC) of 35 seconds (the elapsed time from injection to discharge) starts. During that time, the capillary must be moved and aligned to the current sample column. Once the TC ends, collection of fractions starts. Posteriorly, the capillary is placed in its rest position and the optical fiber is taken. After that, the optical fiber is positioned above each well and EEM are recorded. Finally, the optical fiber is placed in its rest position.

When using TC commands, all tasks inside them must be previously performed in some trials, so as to obtain an approximation of the needed time. Then, the coordinated time must be set to the approximated time plus a few seconds. That excess guarantees the same time intervals for a group of tasks, independently of factors that could affect execution speed. Such control of time is especially important in kinetics experiments. For this reason, several TC are programmed in the Kinetics-EEM sequence (Scheme S-1), so that the needed time for both irradiation and EEM registering are the same for different samples.

Additionally, movies S-1 and S-2 show both sequences during execution.

PARAFAC modeling

Neither data pre-treatment nor constraints were imposed when modeling. For both systems, the number of components for models was three, which was determined on the basis of the knowledge about the real composition of all samples and the spectroscopic activity of compounds. Individual samples were represented by third order tensors ("cubes") composed of a set of EEM that change over time. Information regarding the modeled samples can be obtained from Supporting Information.

RESULTS AND DISCUSSION

Modeling and calibration results

PARAFAC resolved profiles and pure spectra are shown in Figure 1, in which excitation and emission spectra of the pure standards of each analyte were extracted from the respective calibration data matrix which had shown the highest signal value at specific $\lambda_{EX}/\lambda_{EM}$ (Ciprofloxacin-CPF 280/454 nm; Danofloxacin-DNF 285/446 nm; Ofloxacin-OFL 295/496 nm and Azinphos-methyl-AZM 230/395 nm, respectively). As it can be seen, the pure spectra are in good agreement with the profiles obtained from PARAFAC for test samples. In the case of the HPLC-EEM system, spectral profiles for both test samples match the pure spectra satisfactorily, as it can be appreciated from the superposition of each mark (cross and plus) forming an asterisk. Time profiles do not show a classical smoothed chromatographic shape because of the discretization process associated with the joining of drops in the same well. Beyond that, given the fact that those profiles are normalized, in an ideal case all samples should have the same time profiles for modeled components, independently of their concentrations. For example, looking at well number 2 (see details in attached zoom box), OFL points (one for each test sample, and one for each pure analyte calibration sample) show certain dispersion, which could be due to time shifts in the HPLC, or wrong dispensing of drops, among other reasons. Since test samples had three analytes and both green DNF and blue CPF curves do not show dispersion in well number 2, it can be considered that these problems did not occur.

On the side of the Kinetics-EEM system, there is also good correlation in terms of spectral profiles between test and calibration samples. Time profiles reveal the constant level of potential interferences (Thiabendazole-TBZ and Fuberidazole-FBZ), which are not susceptible to light irradiation, and the evolution of AZM through its reaction product. It is worthwhile mentioning that, as was previously stated,[13] AZM have weak native fluorescence.

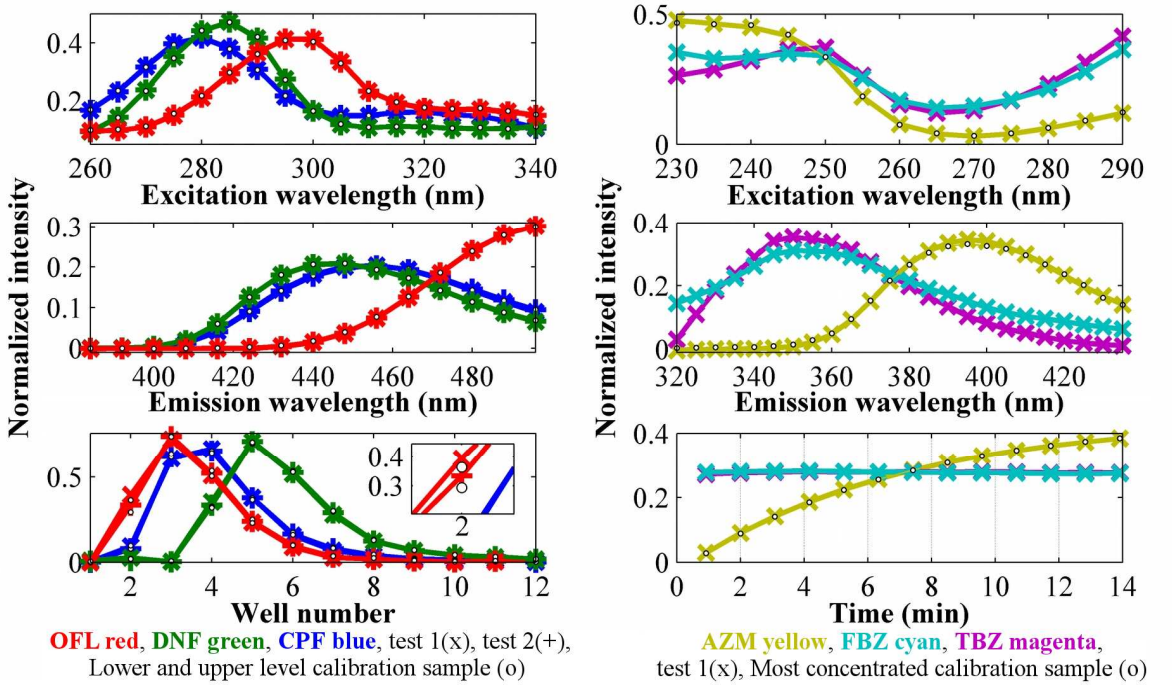


Figure 1. Excitation (top), Emission (middle) and Time (bottom) profiles after PARAFAC modeling of HPLC-EEM (left) and Kinetics-EEM (right) systems.

For both systems, good results in predictions, with recoveries near 100% and low REP% values, were appreciated. For the Kinetics-EEM system, figures of merit are in the same order of magnitude if compared with the previously informed ones. On the HPLC-EEM side, a similar comparison is difficult to attain, since the studied ranges of concentrations have been very different, and also because neither the fluorimeters nor the length of the used optical fibers have been the same. It should be stated that light transmission through optical fibers generally results in a loss of sensitivity, so improvements could be expected with either higher quality or shorter optical fibers. Figures of merit are reported in Table S-1.

Advantages regarding intervention of the operators

In the HPLC-EEM system, since both fraction collection and fluorescence readings were carried out over the 3D Printer using just one plate, there was neither necessity for changing the plate on the collector nor moving it to the plate reader. Once all samples vials were placed in the HPLC, then no more interventions from the operator were required. In case one well plate is not enough, a change of well plate is mandatory, which can be automated too.

Regarding the Kinetics-EEM system, for each sample, the operator only have to wash the cuvette, charge it with 2.00 mL of sample, insert a PTFE stirring magnet and close the fluorimeter compartment. The addition of NaOH, the stirring of the sample in the cuvette, the irradiation cycles and the fluorescence readings were all automated. In the reported work,[13] the operator had to take care about all timing concerns. In this work, the registration of the current time at different points of the sequence allowed us to calculate time dispersion for the same tasks given different samples. As an example, the whole processing for each sample took 825.3 +/- 0.3 seconds. Interestingly, taken into account the high quantity of tasks that had to be performed, reaching a dispersion value near 300 milliseconds would be really difficult to achieve by an operator doing all the required actions manually.

In general, a lesser intervention of the operators could be associated with a higher quality in figures of merit.

Advantages on lesser funds necessity

Regarding the previously reported HPLC-EEM system,[12] here neither fraction collector nor plate reader were necessary. Considering the low cost of the needed components to assemble a 3D Printer and the costs of both a fraction collector and a plate reader compatible with the used HPLC (although the solution reached here does both things with the same system), it was estimated a reduction of costs between 90% and 95% over the prices provided from commercial distributor of equipment.

In the case of the Kinetics-EEM system, since some accessories were specifically designed, it is hard to estimate savings, although it can be approximated as stated in the literature.[26]

Both systems required a holder for optical fibers. The developed one worked appropriately, and also has stirring capabilities, which represents more savings.

The value of D.I.O.S. would be even more difficult to approximate, since it is a program in constant development, so results from future experiments will give real value to this contribution. Last, the chosen license is in agreement with the idea that developers' motivations and beliefs in FOSS principles and redistribution rights may influence license selection more than economic incentive.[9]

General advantages

Any kind of GUI, including those outside the analytical chemistry field, can be virtually handled by means of D.I.O.S. Also, the cross-platform (GNU/Linux and Windows) interface of D.I.O.S. allows users to design sequences without knowledge of programming languages or electronic details.

Regarding the use of 3D printed pieces, their low cost, the possibility of easily design and share the models, and the feasibility of using them to solve specific problems, represent extraordinary advantages. Additionally, when printed pieces are assembled in units comprising electromechanical and electronic components, they can be controlled by Arduino (or similar) boards, which results in an enormous potential when building specific devices, such as customizable 3D Printers. As stated in the literature, analytical chemists usually do not have formal training in electronic engineering.[27] Thus, usage of Arduino boards seems to be a good starting point, since they are designed for others to build and modify them, including documentation and examples of code, besides its low cost and worldwide availability.

CONCLUSIONS

Control of interactions with GUIs of equipment, by means of image recognition tools, represents a powerful strategy to reach certain levels of automation. When these tools are applied through graphical and

intuitive programming practices, those levels can be reached even by scientists without experience in programming or electronics.

Hyphenation/coordination of equipment can be achieved by means of a proper Server-Clients structure, which can be boosted incorporating advantages derived from image recognition tools, and mixing them with generic programming tools (such as loops commands), timing and typing tools, among others. Independently of having a virtual representation, when interactions are also physical ones, a generic CNC machine can be adapted to automate handling of matter and energy.

When all previously described tools are joined in the same system, the experimenter can design automated experimental procedures under a general framework in which equipment, as a combination of virtual and physical inputs and outputs, can be thought as a unit in a controlled and coordinated network.

It was successfully demonstrated that D.I.O.S. meets all the described characteristics. Its usage was exemplified in the context of chemical analysis including third-order data generation Even when it was tested in kinetics and “fraction collection” fluorescence experiments, it can be concluded that similar principles would apply to other types of automations.

Finally, it was verified that interventions of operators can be diminished, which may lead to improvements in terms of precision among repeated actions. In addition, funds savings were substantiated in different ways, for instance, avoiding purchase of hardware and licenses. These savings are derived from free equipment control and coordination through software (D.I.O.S. framework), low cost 3D printed pieces (with addition of some basic materials), open-source electronic platforms, such as Arduino, and projects that can be derived from them, such as RepRap. Regarding the speed, it can be remarked that it depends on the characteristics of the methods (kinetic rates, chromatographic run time, scan speed, among others) and do not strictly depends on the operating way (manual or automatic).

ASSOCIATED CONTENT

Supporting Information

HPLC-EEM system adaptations; Kinetics-EEM system adaptations; Kinetics-EEM sequence; Calibration and Test Samples information; Figures of merit after PARAFAC modeling of both systems; 3D printed components; D.I.O.S. Server GUI, D.I.O.S. Client GUI (**PDF**);

Movie S-1: HPLC-EEM Sequence during execution (**MPEG**)

Movie S-2: Kinetics-EEM Sequence during execution (**MPEG**)

AUTHOR INFORMATION

Corresponding Authors

*E-mail: gabrielsiano@gmail.com and hgoico@fbcb.unl.edu.ar Phone: +54 342 4575206x190.

Author Contributions

All authors have given approval to the final version of the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

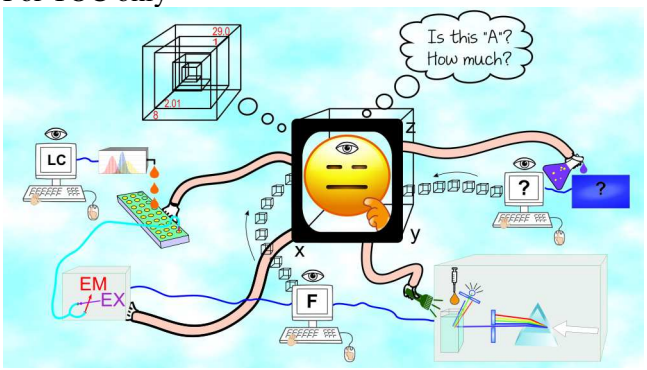
The authors express their gratitude to CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas, Project PIP-2015 Nº 0111) and ANPCyT (Agencia Nacional de Promoción Científica y Tecnológica, Project PICT 2014-0347) for financially supporting this work. M.R.A and M.M gratefully acknowledge the postdoc and Ph.D. financial support, respectively, provided by CONICET. G.G.S. thanks Gabriel Gómez, Juan Carrique and Ignacio Elcoro for their support.

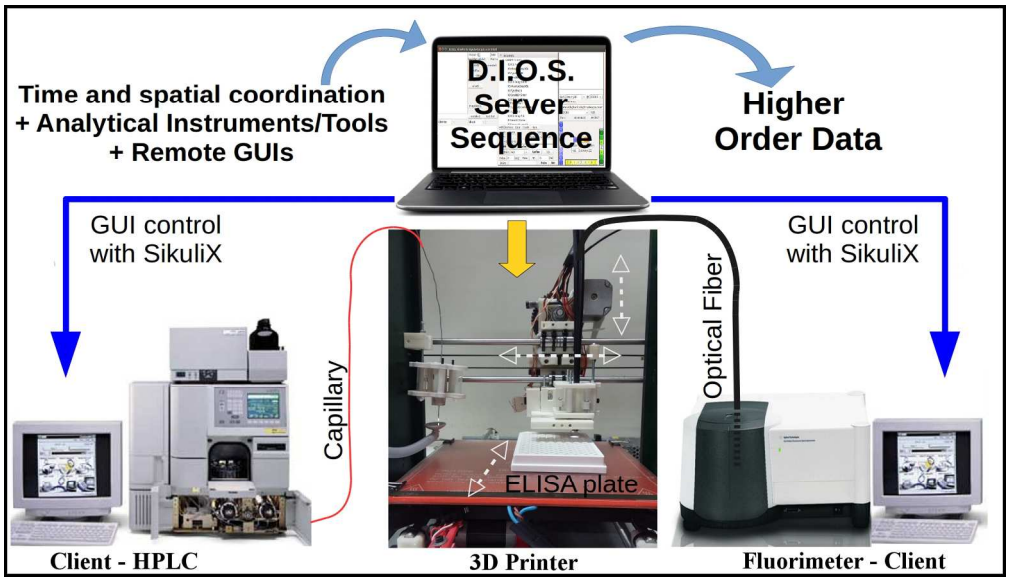
REFERENCES

(1) Maggio, R. M.; Damiani, P. C.; Olivieri, A. C. *Anal. Chim. Acta* **2010**, *677*, 97–107.
(2) Qing, X.-D.; Wu, H.-L.; Zhang, X.-H.; Li, Y.; Gu, H.-W.; Yu, R.-Q. *Anal. Chim. Acta* **2015**, *861*, 12–24.
(3) Booksh, K. S.; Kowalski, B. R. *Anal. Chem.* **1994**, *66*, 782A–791A.
(4) Escandar, G. M.; Goicoechea, H. C.; Muñoz de la Peña, A.; Olivieri, A. C. *Anal. Chim. Acta* **2014**, *806*, 8–26.
(5) Carabajal, M. D.; Arancibia, J. A.; Escandar, G. M. *Talanta* **2017**, *165*, 52–63.
(6) Yeh, T.; Chang, T.-H.; Miller, R. C. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*; UIST '09; ACM: New York, NY, USA, 2009; pp 183–192.

(7) Wijnen, B.; Hunt, E. J.; Anzalone, G. C.; Pearce, J. M. *PLoS ONE* **2014**, *9*, e107216 .
(8) Pearce, J. M. *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*, 1st ed.; Elsevier Science Publishers B.V.: Amsterdam, 2013.
(9) Dryden, M. D. M.; Fobel, R.; Fobel, C.; Wheeler, A. R. *Anal. Chem.* **2017**, *89*, 4330–4338.
(10) Arduino - Home https://www.arduino.cc/ (accessed 2017/03/20).
(11) Jones, R.; Haufe, P.; Sells, E.; Iravani, P.; Olliver, V.; Palmer, C.; Bowyer, A. *Robotica* **2011**, *29*, 177–191.
(12) Alcaráz, M. R.; Siano, G. G.; Culzoni, M. J.; Muñoz de la Peña, A.; Goicoechea, H. C. *Anal. Chim. Acta* **2014**, *809*, 37–46.
(13) Montemurro, M.; Siano, G. G.; Culzoni, M. J.; Goicoechea, H. C. *Sens. Actuators B Chem.* **2017**, *239*, 397–404.
(14) Prusa, Josef. Prusa i3 3D printer http://www.prusaprinters.org/prusa-i3/ (accessed 2017/03/20).
(15) Van der Zalm, E; Gubbels, C. *Marlin*; Marlin Open Source 3D Printer Firmware, version 1.0.0; 2011.
(16) Python Software Foundation. *Python*; Python Language Reference, version 2.7.
(17) Bro, R. *Chemom. Intell. Lab. Syst.* **1997**, *38*, 149–171.
(18) Olivieri, A. C.; Wu, H.-L.; Yu, R.-Q. *Chemom. Intell. Lab. Syst.* **2012**, *116*, 9–16.
(19) The MathWorks. *MATLAB - The Language Of Technical Computing, r2010a*; Natick, Massachusetts, U.S., 2010.
(20) Raimund Hocke. *SikuliX*; version 1.1, http://sikulix.com/.
(21) JetBrains. *PyCharm*; PyCharm Community Edition, version 2016.2.
(22) Google. *SketchUp*; Google SketchUp, version 2013; 2013.
(23) Ultimaker. *Cura*; Cura Software, version 15.04.6; 2016.
(24) kliment - GitHub. *Printrun*; Pronterface, Pronsole, and Printcore - Pure Python 3d printing host software, version 20140406.
(25) The GNU Affero General Public License. https://www.gnu.org/licenses/agpl-3.0.en.html (accessed 2017/03/20).
(26) Pearce, J. M. *Mod. Econ.* **2015**, *06*, 1.
(27) Urban, P. L. *The Analyst* **2015**, *140*, 963–975

For TOC only





Scheme 1. D.I.O.S. features

177x101mm (300 x 300 DPI)

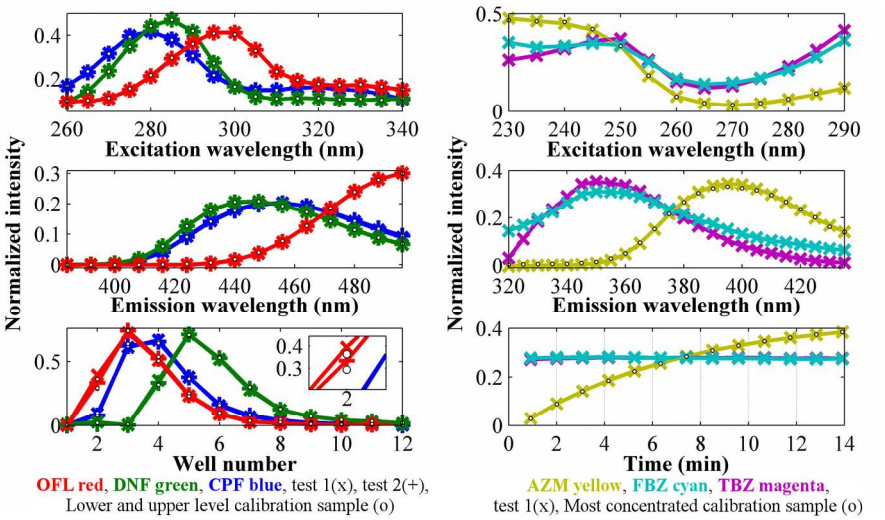


Figure 1. Excitation (top), Emission (middle) and Time (bottom) profiles after PARAFAC modeling of HPLC-EEM (left) and Kinetics-EEM (right) systems

177x91mm (300 x 300 DPI)

```
Loop samples 1:8
  G-code: Take CT and wait above drain funnel
  Sikuli: HPLC GUI: set number of sample (vial);
  request injection; wait until GUI confirms injection
  Start Time Coordination (35")
  G-code: move CT, align it with respective plate column
  Finish Time Coordination (35"): Wait if needed
  G-code: move CT to the first well, stay 1.5" and repeat
  for wells in the same column (11); hang CT; Take OFT
  Loop well 1:12
    G-code: move OFT above a well
    Sikuli: Fluorimeter GUI: request new matrix scan
  G-code: hang OFT
Server,Printer,Client,CT:Capillary Tool,OFT:Optical Fiber Tool
```

Scheme 2. D.I.O.S. Sequence for HPLC-EEM system (pseudo-code)

84x55mm (300 x 300 DPI)