

CocoSoft: educational software for automation in the analytical chemistry laboratory

David J. Cocovi-Solberg¹ · Manuel Miró¹

Published online: 5 July 2015
© Springer-Verlag Berlin Heidelberg 2015

Introduction

The implementation of European Credit Transfer System (ECTS)-based majors and subject courses has launched a vast number of innovative teaching and learning methods over the recent past [1–5]. Lecturers and students are currently bearing new roles compared with traditional syllabuses in Higher Education Institutions. The lecturer is not tagged anymore as the active and sole element within the knowledge transfer chain in ECTS disciplines, yet the vehicle for continuous generation of active knowledge. Students are expected to possess a new role as the actual managers of their own learning in a formative assessment framework [6]. Undergraduate laboratory courses in the analytical chemistry syllabus such as ‘Advanced chemistry laboratory’ or ‘Integrated chemistry laboratory’, to name a few, are now focused on solving real-life scenarios involving chemical tools in which students in a guided-inquiry format shall choose by themselves the analytical techniques that better fit their purpose, learn the fundamentals and basic operational principles of those techniques, and acquire knowledge on how to process experimental data correctly from the mathematical and chemical view points on their own [7]. The limited number of lecturers allocated to every single laboratory exercise and the increasing workload of the ECTS-based

analytical chemistry syllabus make these learning objectives become challenging tasks to accomplish. As a result, the course learning objectives are increasingly common calling for straightforward unit operations (e.g., the loading of a preprocessed sample aliquot in a given analyzer) but students might not be aware of the various components of the analyzer and how the instrumental setting is controlled by a PC, thus lacking the appropriate knowledge to face and track failures in analytical instrumentation, or how to improve the analytical procedure outside the range of predefined experimental conditions. Chemistry students are trained to introduce the experimental data into a macro spreadsheet or statistical software, but they are barely able to understand what algorithms are behind the scenes. An attempt to teach the underlying software, electronics, communications, and mechanics of instrumental analyzers from engineering viewpoints might greatly improve student proficiency in the analytical chemistry laboratory curriculum.

At the postgraduate level, chemometric tools for visualizing relevant figures from datasets are usually well understood by students, but the increasingly larger datasets are making the tracking of errors cumbersome. Courses focused on laboratory control, automation, or data processing are in many instances obsolete, unspecific, and lack real ‘hands on’ experiments on programming topics, and are merely dedicated to solve focused tasks in ad hoc short sessions, but with difficult adaptation to a broad set of real-life scenarios that are expected in ECTS-based disciplines.

There is actually a quest of open source software packages for basic laboratory control and data processing designed in user-friendly and educational format, in which the algorithms used to solve a given case study might be read directly from the font code, assisting students in acquiring proficiency in the analytical chemistry laboratory. Commercial software packages regrettably are barely free or open. They tend to be

Electronic supplementary material The online version of this article (doi:10.1007/s00216-015-8834-8) contains supplementary material, which is available to authorized users.

✉ David J. Cocovi-Solberg
dj.cocovi.solberg@gmail.com

¹ FI-TRACE group, Department of Chemistry, University of the Balearic Islands, Carretera de Valldemossa, km 7.5, E-07122 Palma de Mallorca, Spain

dedicated to a handful of instruments or execute pre-set operations rather than automating the entire analytical process. As a consequence, payware works as a black box, in which the machinery cannot be seen and potential errors are difficult to trace back by users. To understand the math or computing behind the experimental results (or errors), this type of software package is worthless. More importantly, commercially available software packages are expensive, restrain some functions the hardware could do originally and, in some cases, distinct individual software packages are needed to operate various analytical instruments. Experimental data must usually be exported several times in a humdrum and time-consuming manner that prevents users from getting new data or processing results in a timely manner. Default commercial unsupervised algorithms lead to errors, and customized algorithms can only be implemented in customized software packages, not in proprietary ones, so data treatment is not as automatic as it could be. Several approaches toward dedicated software packages for the control of instrumental equipment and/or data acquisition have been published over the past few years. Notwithstanding the fact that the reported software packages are open-source and allow for a ‘hands on’ startup, they are still deemed to be directed to specific hardware [8–12], rely upon predefined libraries [13], and even if algorithms or communication protocols are known, the addition of new hardware is not straightforward and requires manufacturer’s assistance [14–16]. Available open-source software packages also bear limited options for data processing, making the implementation of feedback protocols in analytical methods cumbersome, jeopardizing the development of fully automated smart methods.

In this contribution, a novel user-friendly and open educational software package, so-called Cocosoft, is proposed for automation of analytical procedures, including instrumental control, data treatment, and smart method development, in which the font code can be either readily read to get insight into the algorithm or added by the user for controlling new hardware or executing new data processing procedures. This software package can also serve as a valuable educational tool for introducing coding exercises at the graduate and postgraduate course level so as to trigger problem resolution capabilities of the students as expected in ECTS-based subject matters.

Cocosoft features

The software package presented herein allows for facile automation of analytical procedures and control of instrumentation as endorsed in the analytical chemistry laboratory curriculum with the additional advantage of processing experimental data at real time, all aided by customized mathematical procedures. It is written in Python programming language. This is a

multiplatform that has proven to be extremely readable against other engineering-directed software packages, such as Matlab or Labview [17]. Cocosoft is composed of a friendly graphic user interface (GUI), a file management system, and a tool for highlighting of the script that allows users to visualize the analytical procedure step under execution at real time. A close-up of the software main window and components thereof is given in Fig. 1.

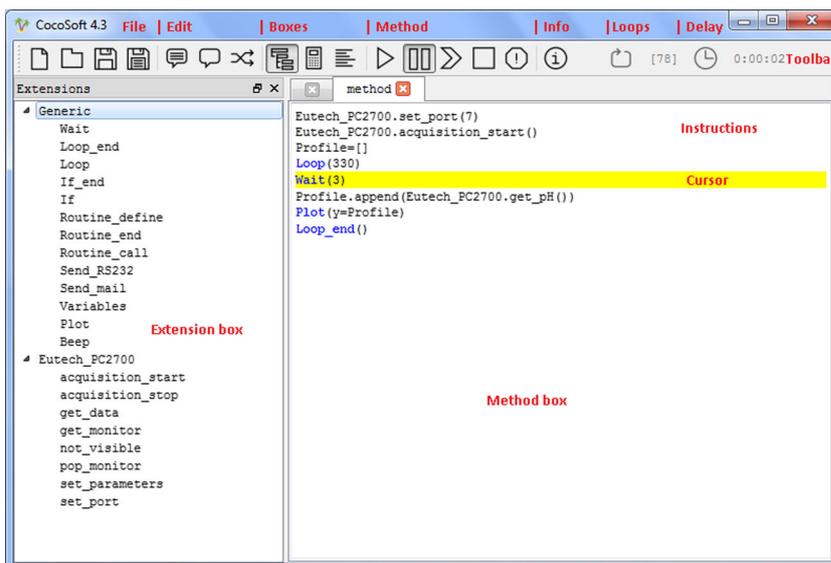
Any hardware to be controlled by Cocosoft should have a related extension (Python script) that contains all necessary functions to successfully control it. Any python file (‘.py’) that is in the same folder than that of Cocosoft will be recognized as an extension, and its functions imported and available from GUI with a single mouse click. Any native Python syntax is accepted as instruction in the program, thus expanding Cocosoft’s functionality far beyond extensions. The program executes the entire set of steps of the analytical procedure in a one-at-a-time format so as to wait for the instrument or function feedback before pursuing the next instruction. Failures related to inaccurate hardware timing are thus avoided. This is in fact seen as the main drawback of other custom-made software packages [8].

Font code is comprehensively commented and accessible with any text processor. Students are entitled to scrutinize the overall operation of the program and the syntax of any instruction in the analytical procedure. Cocosoft user guide, attached as [Electronic Supplementary Information](#), comprehensively describes the GUI, the flow diagrams of the program, several method examples, and tips and tricks for coding custom extensions for analytical instruments, analytical procedures, and mathematical operations.

Cocosoft tutorial in undergraduate courses

In order to follow the guidelines set by the European Higher Education Area (EHEA) regarding the use of problem-based learning in the analytical chemistry laboratory curriculum at the undergraduate level, an interpretative lesson about the relevance of monitoring and maintaining the extractant pH in bioaccessibility tests of trace elements in soil environments—harnessing the single-extraction procedure as endorsed by the EU Standards, Measurement, and Testing (SM&T) program using 0.43 mol/L AcOH as leaching medium—was taken as a tutorial example. The idea behind is to supplement laboratory assays combining acid-base reaction tests with potentiometric detection. Based upon constructivism active learning, students should observe the increase of extractant pH upon dissolution of soil carbonates whereupon they should design appropriate analytical procedures supported by software algorithms to sustain the nominal extractant pH for reliable measurements throughout the extraction test, which, according to SM&T, lasts 16 h.

Fig. 1 Screenshot of CocoSoft main window in Windows 7 illustrating the method script and extensions available. The toolbar includes file management, edition of the method, method flow, information, loops, and delay icons



From an educational viewpoint, the pH-stat methodology [18] is to be described first, namely, the addition of minute aliquots of a strong acid to the extraction medium to compensate for the neutralization of the acetic acid by dissolved carbonates, followed by the presentation of the hardware and software employed in the tutorial example: (i) Eutech PC2700 pH-meter, (ii) CAVRO XP3000 syringe pump, (iii) magnetic stirrer, (iv) extraction vessel, (v) 3 mol/L HCl reservoir, and (vi) CocoSoft for controlling all of the instrumentation. Figure 2 shows a diagrammatic description of hardware connections and the information flow path between hardware and Cocosoft.

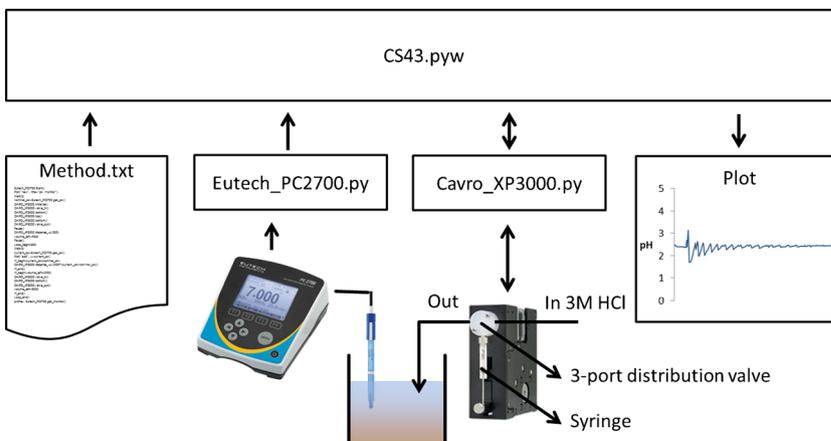
A monitoring algorithm is first taught and provided to the students. This is followed by running the standard SM&T method for 15 min while describing the main pitfalls that could stem from the gradual increase of the extractant pH (e.g., the lower extracting capability of the acetic acid and

the underestimation of the real hazard of contaminated soils by trace elements) The SM&T single extraction test was executed by adding 2.0 g of a calcareous agricultural soil (Palma de Mallorca, Spain) to 80 mL of 0.43 mol/L CH₃COOH subjected to controlled mechanical agitation (500 rpm) throughout. A simple Cocosoft script is given below as an educational example:

```
Profile=[]
Loop(330)
Wait(3)
Profile.append(Eutech_PC2700.get_pH())
Plot(y=Profile)
Loop_end()
```

The role of every single instruction in the automatic analytical method above for continuous monitoring of the pH is to

Fig. 2 Conceptual diagram of the distinct instrumental components and connections used in this tutorial. The pH-meter and the syringe pump are connected to the PC through USB-RS232 converter. The main CocoSoft program controls hardware components via their extensions. A .txt file containing the method is executed automatically. Data is read continuously from the pH-meter and triggers the motion of the syringe pump in due time. The pH profile is plotted in the screen



be explained, including setting up of the communications, data acquisition, and data plotting. An illustrative profile of pH monitoring is shown in Fig. 3A.

Upon critical evaluation of the pH profile, students should be asked to search possibilities and write viable scripts for automatic addition of HCl aliquots to the extraction vessel. They are distributed in small groups to trigger collaborative work in line with EHEA recommendations. The lecturer should indicate a number of instructions available of the CAVRO XP3000 pump while providing tips and tricks of what could be done with the equipment assembled.

The prevailing answer expected is to add fixed-minute volumes of HCl at predefined times aided by the syringe pump. The amount to be added, however, varies from soil to soil, and although this protocol is recommended in the standard CEN/TS 14429:2005, as students have access to smart automatic equipment, no further attention was given to this method. Emphasis was placed on the word 'smart', and on the possibility to further improve the pH monitoring system if we succeed in programming a viable automatic method with feedback from the sample.

After further discussion in a team collaborative environment, the possibility of design of a pH comparison system should come up. This was the case in a last year undergraduate course of advanced analytical chemistry laboratory in which this tutorial example was introduced as an innovative learning tool. The basis of the comparison algorithm is as follows: 'when the extract pH is higher than the nominal pH of the extractant, a given volume of acid should be added (in our case 200 μL (see script below)'. The fluidic system was initialized manually, and four instructions were added to the previous monitoring script:

Before the loop:

```
Initial_pH = Eutech_PC2700.get_pH()
```

And in the loop, before the plotting:

```
If(Eutech_PC2700.get_pH(>Initial_pH)
CAVRO_XP3000.dispense_uL(200)
If_end()
```

The time-based pH profile of the SM&T test was projected at real time in the class (see Fig. 3B) and synchronized with the visualization of the Cocosoft script. This test served in triggering students' learning outcomes and engagement as a great expectation was observed after the first addition of acid in as much as most students were eagerly waiting for the pH to

exceed the nominal extractant pH for the automatic actuation of the syringe pump.

Several teams, however, soon realized that the main drawback of the smart comparative procedure is that once the carbonate pool of the solid material is almost entirely dissolved in the acetic acid milieu, occurring within a few-minute timeframe, the difference of 100th of a pH unit still triggered the addition of an aliquot of 200 μL of 3 mol/L HCl with the consequent sharp decrease in the extractant pH, which is not to be buffered. The lecturer should make students aware that the dissolution of further mineralogical phases that have slower leaching kinetics will be carried out at lower pH values, which will in turn overestimate the pools of bioaccessible trace elements in risk assessment studies of metal contaminated soils. As a result, the students in the laboratory course should identify the need of a different algorithm to tackle this issue. Our experience indicates that at this point the educator should give some further tips and tricks prior to introducing the so-called proportional algorithm in which the volume of the acid aliquot added to the extraction medium is proportional to the difference between the nominal and current pH values rather than adding a steady volume regardless of the absolute pH value, as is the case with the comparator algorithm. The procedure is the same as in the previous example, but the smart conditional line should be replaced from:

```
CAVRO_XP3000.dispense_uL (200)
```

to:

```
CAVRO_XP3000.dispense_uL (1000*(Eutech_PC2700.get_pH()-Initial_pH))
```

This demonstrates the versatility of the Cocosoft software package in method development and optimization of analytical procedures. With the proportional algorithm, two simple demonstrations could be introduced to illustrate the fact that a difference of one pH unit will lead to the addition of a 1000 μL acid aliquot to the extraction vessel, whereas a difference of 100th of a pH unit will trigger the automatic addition of a mere 10 μL aliquot. The new monitoring profile (proportional algorithm) is compared in Fig. 3C with that of the previous procedure (comparator algorithm).

As a further team collaborative work, students might be asked to enumerate potential improvements to the algorithm. We do expect that they envisage the feasibility of the first derivative for monitoring of the rate of pH change. Optimization of the parameters of the proportional algorithm, including the time between consecutive measurements and the constant of proportionality

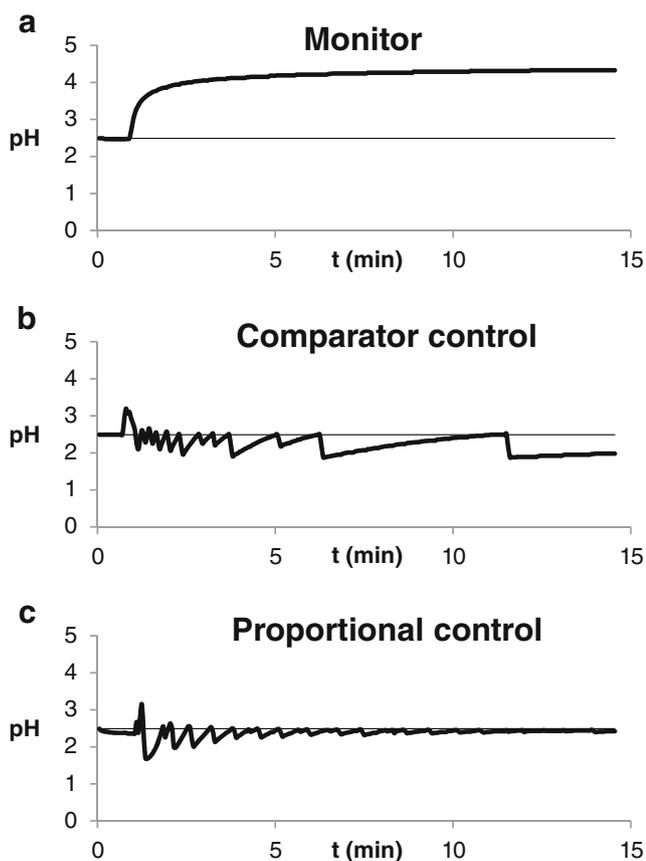


Fig. 3 Extraction profiles illustrating the changes of pH values as obtained with (A) monitoring test, (B) comparator algorithm, and (C) proportional algorithm in a 15 min-extraction timeframe

between the acid volume and the difference between the nominal and actual pH values, needs also to be perceived and illustrated via practical examples in the laboratory.

Cocosoft tutorial in postgraduate courses

Cocosoft was also introduced in advanced courses of automation in the analytical chemistry laboratory addressed to Master and PhD students so as to set up computer-controlled fluidic manifolds composed of pressure-driven liquid drivers, sample injection systems, gas-diffusion units, and potentiometric detectors for characterization of agricultural soils in terms of ammonium content [19]. Lecturers should, in a tutorial format, comprehensively pinpoint Cocosoft GUI main components and advanced functions thereof (see Fig. 1 for details). Instructions for writing a basic script serving as analytical protocol should be given to students for interactively

learning the underlying principles of programming and features of Cocosoft in the analytical laboratory for operation of syringe pumps, rotary valves, and measurement and processing of potentiometric data from ion selective electrodes. Students should afterwards be actively engaged in writing new analytical procedures using an emerging educational model of laboratory-based formative assessment on the basis of the initial analytical protocol presented before to, for example, move the valve from port 3 to port 4 or wait 30 s rather than 3 s for efficient permeation of evolved ammonia in the flow system. In both cases, students should realize that a change of the number in brackets would suffice for proper script modification. At this point, lecturer–student interaction is deemed imperative for building a fully automated analytical procedure for determination of ammonium in soil extracts in a flow-based configuration incorporating gas-diffusion separation and understanding the underlying acid-base reactions involved. The lecturer does take the lead here but requires students’ feedback on how to code the different instructions. Students are also encouraged to take notes on more specific challenging tasks, for example, replace the communication port and execution of optimization protocols of method parameters such as flow rates, sample/reagent volumes, and delays by straightforward modification of numerical values in the Cocosoft script. In this way, students acquire proficiency in method programming on their own. At this point, we have observed that most postgraduate students merely perform minor changes in the arguments of the instructions, but do not dare writing new instructions. After more than 1 wk of full-time “hand-on” training on software functionalities taking the flow-based potentiometric system as a tutorial example, we have seen that postgraduate students succeeded in making analytical methods from scratch efficiently by copying groups of instructions of defined functionality from previous methods and incorporating unusual instructions from available extensions.

Conclusion and outlook

In this educational contribution, Cocosoft is proposed as a versatile educational software package for control and data processing of all sorts of basic instrumentation in advanced analytical chemistry laboratories at the undergraduate and postgraduate levels to cope with the EHEA specifications for the chemistry syllabus. This open source software is written in Python programming

language and contains the main program and extensions thereof that can be added by every user without a comprehensive knowledge of Python. A short trial- and-error period is deemed sufficient for bachelor and post-graduate students to grasp the fundamental principles and functionalities of the program so as to build custom extensions on their own for communication with a vast number of analytical instruments and processing of complex datasets as per EHEA guidelines on student's self-learning abilities. Beside its educational features, the software is fully functional as a laboratory automation and data treatment tool.

Future work faces two clear fronts. On one side, the incorporation of new features to the basic Cocosoft code in terms of ancillary functions, data processing, statistics (namely, data smoothing, calibration, optimization schemes), and the overall GUI aspect. Research work is also underway in proxying mouse and keyboard patterns without human intervention, aiming at setting fully automated smart methods exploiting modern analytical equipments (e.g., chromatographs, atomic absorption or emission spectrometers, mass spectrometers, or nuclear magnetic resonance spectrometers), which do not usually offer the option to be controlled by software other than that of the manufacturer. On the other side, different extensions are now being developed for a plethora of fluidic devices and detectors of a variety of manufacturers, (e.g., Crison, Fialab, Emstat, Ocean Optics, CMA, UNI-T, AIM, Ontrak, IDEX, Ismatec, and Hamamatsu, to name a few).

Open software packages, as demonstrated herein by Cocosoft, have great capabilities in the educational and the analytical chemistry research arenas as fully functional controllers for analytical instrumentation, but it is virtually unfeasible to code extensions for every existing instrument. To tackle this issue, sharing extensions of instruments and mathematical procedures, coded by different users or students, will allow for a fast and robust network development.

Acknowledgments The financial support from the Spanish Ministry of Economy and Competitiveness through projects CTM2014-56628-C3-3-R and CTM2014-61553-EXP, and allocation of a PhD stipendium to D.J.C-S is gratefully acknowledged. The authors express their gratitude to Maria E. Rosende, Javier Llorente Downes, Luis M. Laglera, and Daniel Kremr for their suggestions in improving Cocosoft aesthetics and functionality.

References

- Dabbagh N, Bannan-Ritland B (2005) Online learning: concepts, strategies, and application. Pearson Prentice Hall, New Jersey
- Kozaris IA (2010) Platforms for e-learning. *Anal Bioanal Chem* 397:893–898
- Garrison DR, Vaughan ND (2008) Blended learning in higher education: framework, principles, and guidelines. Jossey-Bass, San Francisco
- Garrison DR, Kanuka H (2004) Blended learning: uncovering its transformative potential in higher education. *Internet High Educ* 7: 95–105
- Cole J, Foster H (2008) Using Moodle: teaching with the popular open source course management system, 2nd edn. O'Reilly Media Inc., Sebastopol
- Heather F, Ketteridge S, Stephanie M (2008) A handbook for teaching and learning in higher education: enhancing academic practice, 3rd edn. Routledge, New York
- Fakayode SO (2014) Guided-inquiry laboratory experiments in the analytical chemistry laboratory curriculum. *Anal Bioanal Chem* 406:1267–1271
- Knochen M, Caamaño A, Bentos H (2011) Development of a low-cost SIA-based analyser for water samples. *J Autom Methods Manage Chem* 2011, 943465. doi:10.1155/2011/943465
- Skelarova H, Svoboda A, Solich P, Polasek M, Karlicek R (2002) Simple laboratory-made automated sequential injection analysis (SIA) device. II. SIA operational software based on Labview (R) programming language. *Instrum Sci Technol* 30:353–360
- Dominguez R, Muñoz R, Araiza H (2010) Automated analytical system based on the SIA technique, Pan American Health Care Exchanges, Available at: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5474585>. Last Accessed on 2 May 2015
- Barzin R, Shukor S, Ahmad A (2010) New spectrophotometric measurement method for process control of miniaturized intensified systems. *Sensors Actuators B Chem* 146: 403–409
- Jitmanee K, Teshima N, Sakai T, Grudpan K (2007) ICP-MS coupled with automated flow injection system with anion exchange minicolumns for determination of selenium compounds in water samples. *Talanta* 73:352–357
- Wagner C, Genner A, Ramer G, Lendl B (2011) Advanced Total Lab Automation System (ATLAS), modeling, programming and simulations using LabVIEW™ software. In: De Asmundis R (ed) *InTech*, Available at: <http://cdn.intechopen.com/pdfs-wm/12934.pdf>. Accessed 2 Mar 2015
- Becerra E, Cladera A, Cerda V (1999) Design of a very versatile software program for automating analytical methods. *Lab Robot Autom* 11:131–140
- Information on Fialab for windows and FIAsoft software, Available at: <http://www.flowinjection.com/software/>. Accessed 2 May 2015
- Information and downloads of FloZF software Available at: <http://www.globalfia.com/software/flozf>. Accessed 2 May 2015
- Fangohr H (2004) A comparison of C, MATLAB, and Python as teaching languages in engineering. *Lect Notes Comput Sci* 3039: 1210–1217
- Rigol A, Mateu J, González-Núñez R, Rauret G, Vidal M (2009) pH(stat) vs. single extraction tests to evaluate heavy metals and arsenic leachability in environmental samples. *Anal Chim Acta* 632:69–79
- Ferreira AMR, Lima JLFC, Rangel AOSS (1996) Potentiometric determination of total nitrogen in soils by flow injection analysis with a gas-diffusion unit. *Aust J Soil Res* 34:503–510



David Cocovi-Solberg is a PhD student in the group FI-TRACE of the Department of Chemistry at the University of the Balearic Islands. His research focuses on developing automatic methods for assessing bioaccessibility of pollutants and nutrients in environmental solid samples with particular interest in electronics, software, and fluidic components of the analytical setups. He is also proficient in chromatographic approaches and statistics.



Manuel Miró is currently Associate Professor of Analytical Chemistry at the University of the Balearic Islands. He is also a member of the IUPAC subcommittee on Chemical and Biophysical Processes in the Environment and has authored 140 refereed articles including nine book chapters, with an h-index of 31 and over 2500 citations. He was awarded the Flow Injection Analysis (FIA) Medal for Science (2013) by the Japanese Association for Flow Injection Analysis.

His research interests are focused on the development of on-line sample processing strategies for isolation and/or preconcentration of trace levels of environmental pollutants exploiting the various generations of flow injection, including lab-on-a-valve mesofluidic platforms, in hyphenation with modern analytical instrumentation. He is also interested in information and communications technology (ICT) tools for formative assessment in the analytical chemistry curriculum.