EDITORIAL

Pharmacometrics and Systems Pharmacology Software Tutorials and Use: Comments and Guidelines for PSP Contributions

In addition to methodological Tutorials,1 CPT:PSP has recently started to publish software Tutorials.2,3 Our readership and authors may be wondering what kind of format or product is expected, and the review of submissions we have already received prompted several discussions within the PSP Editorial Team. This editorial reflects on these discussions and summarizes their salient points. It aims at providing some details about the current vision of CPT:PSP for software tutorial articles. In addition, it brings some clarity on the topic of what role commercial software tutorials can have in CPT:PSP and how CPT:PSP tutorials differ from publications which describe the software itself, as those which can be found in other computer science journals. Finally, the discussion includes reproducibility considerations and the general use of commercial and noncommercial software in CPT:PSP publications. We hope our thoughts, and especially a stated requirement to publish user input to the software to aid in reproducibility, will help in guiding our authors and will stimulate healthy debate among our readers about the evolving nature of our science, how it can be facilitated using software and associated databases as a conduit, and what role this journal can play in fostering both the best modeling and simulation practices and the best scientific approaches to computational modeling, to bring the advantages of modeling and simulation to all regular practitioners, and not to just a (self) selected few.

CPT: Pharmacometrics & Systems Pharmacology (2013) 2, e86; doi:10.1038/psp.2013.60; published online 18 December 2013

SOFTWARE TUTORIAL VS. USER GUIDE

For the pharmacometrician and the system pharmacologist, computer software is, quite simply, the equivalent of the laboratory for the molecular biologist, or of the research clinic for the practicing clinician. It is the environment where tools are developed, observations and conjectures are made, hypotheses are formulated and tested, multiple scenarios are run, and discoveries are made for the betterment of science or patients. It is only natural that CPT:PSP, in its efforts to merge pharmacometrics and systems pharmaco-ology, wishes to be the home to cutting-edge tutorial papers that can bring practitioners of one discipline closer to the other, spreading beyond the initiated to familiarize our audience with an ever-increasing number of tools. Such familiarity is expected to facilitate new findings, developments, and discoveries by indicating capabilities and potential applications of pharmacometrics and systems pharmacology software and helping with repeatability of the study, an essential element in any scientific discipline. In the words of Umberto Eco, thinking back to another powerful medium used to propagate information, “if you want to use television to teach somebody, you must first teach them how to use television.”

The CPT:PSP designation for a Tutorial, as stated in the journal’s Guide to Authors, is an “educational article providing practical tutorial on tools, methodologies, and approaches in pharmacometrics and systems pharmacology.” This is of crucial importance for the continued health of both pharmacometrics and systems pharmacology because the pace of changes in these dynamic fields appears to outpace any concerted efforts to provide timely didactic needs by current higher education systems. An integrated technical education on modeling and simulation tools has been sporadic, carried out individually in centers scattered around the globe.

As explained in the inaugural Editorial4 and in a description of our disciplines foundational tutorials,1 CPT:PSP strongly supports (technical) tutorials that aid the dissemination and uptake of model-based approaches in pharmacometrics and systems pharmacology, targeting a wide spectrum of audiences from novices to advanced modelers. Inevitably, many of these tutorials may focus on one or more software tools, some of which may be commercial products or otherwise proprietary. The CPT:PSP Editorial Team has been debating the content of the tutorials, their scientific scope, and also whether in any way to limit the scope of such tutorials to those where the software is available for free (whether tout court, for research purposes, or for research purposes to noncommercial organizations only), because this fits into the mission of the journal to support the dissemination of the discipline. This editorial is the result of our deliberations and defines our current position on this topic; we are putting it forth in CPT:PSP to be proactive and in the hope that it will stimulate some needed debate.

An important (but often underappreciated) distinction we need to make early on is between a software tutorial and a user’s manual. We would most often consider user’s manuals not publishable by CPT:PSP, as they consist of a sequence of steps and/or reference material that software users can consult as they solve real-world problems or attend a hands-on workshop to learn the mechanics of the software. CPT:PSP looks to publish software tutorials that contain a real-world scientific problem (case study) and apply the method or the software to solve that problem step-by-step, much like it is done in a real-world scientific inquiry scenario. That way, the usefulness of the document is increased, as the reader now learns twice over: from the scientific basis of the solution to the case study, and the operational aspects of the specific computer software being used. A software tutorial is thus
richer than a user’s manual, as it includes both a description of the scientific problems the software is meant to solve and the practicalities of the software tool itself, whereas a user’s manual most often contains only the latter. We collectively believe that CPT:PSP should focus on software tutorials, whereas user’s manuals may be published separately (in a biomedical computing journal, for example) or become a useful appendix to a tutorial manuscript (e.g., available through a web link). In general, the tutorials should provide a balanced view of the field and include consideration of other, similar software.

ACCESSIBILITY OF SOFTWARE AND MODELING STUDY REPEATABILITY

The commercial nature of some of the tools that we use also bears discussion. PLOS Computational Biology has taken an extreme position on this in a recent Editorial. The journal requires all software to be freely available under the Open Source Initiative for an article to be considered for publication. It is our belief that this may be untenable in our disciplines. The best case in point would probably be that we could not publish a NONMEM tutorial if we were to apply such criteria. To us, there seems to be a fundamental difference between the bioinformatics/computational biology discipline (where software has always been free and open source, since the very early days of the field) and ours (where software has very often, almost exclusively, been commercial). However, unambiguously adjudicating the scientific merit of publications requires that the study be repeatable by other researchers, and this will still be facilitated when the files and data used for any piece of research are made available to others, as opposed to the software platform itself. An analogy can be drawn when details of an analytical method to assay chemicals or genes are given, but the ability to repeat the study is still impinging on access to a liquid chromatography–mass spectrometry machine or gene sequencer, which obviously cannot be easily provided as an Appendix to the report!

GUIDANCE ON PREPARING SOFTWARE TUTORIALS

Let us elaborate further on these aspects and on what this implies for the authors and readers of software tutorials published in CPT:PSP. In many ways, the intent of the more traditional approach of including in research articles the model (differential) equations in plain text ensures that mathematical models can be ultimately implemented (with a bit of work!) in any computer code. However, everyone with an interest in this topic is painfully aware that what gets in the way of the process is that the information is usually incomplete and the textual reporting is error prone. Publishing or making available computer code or modeling scripts is intended to streamline the process of implementing new models described in CPT:PSP research articles. This of course also impacts reproducibility of tutorials. Initiatives have recently been launched to facilitate the introduction of standards and common languages. Curated databases are also increasingly available that are accessible to all free of charge, without significant restrictions. It is apparent that databases such as these are of value, and unrestricted access has in general to be evaluated for costs and benefits against other modes of dissemination. We certainly see availability of computer code as a very important avenue to maximize the impact of our published tutorials.

A compromise that we are already striving to implement in the CPT:PSP research articles would be to upload enough information to enable the reader to reproduce the model results, assuming the software platform is broadly available, perhaps at a cost. This could be as simple as a parameter set (in native format), or as detailed as the actual model code or tutorial script (e.g., control files for NONMEM), workspace file, and so forth, depending on the specifics of the platform used. This would enable the reader access to the software to reproduce the simulations, even if the software is commercial or otherwise proprietary. The only requirement would be that the journal reader needs to be able to run the code in the native software format and reproduce the published simulations (for verification) with no modifications. Of note, this approach is indirectly borne out in the life sciences: a DNA sequence is required to be published in some journals, but this does not require the author to supply a sequencer machine as well. The tools to run a simulation can be separate from the conditions and inputs that need to be specified, i.e., all the details that permit an independent reader re-run the study. This is in line with the requirements from other journals, and it is our belief that such an approach would satisfy the great majority of readers.

MANAGING CONFLICTS

Financial interests are often complex and hard to define, and software fees vary widely in the amount of profit they bring. Some tools may require a fee which includes user support, and others may charge separately for extended service (through consulting, for example). At any rate, we feel that there may be a place in CPT:PSP also for commercial software. This is provided the considerations we outlined above about reproducibility of the simulations can be met. We encourage submissions of tutorials where authors are not exclusively employed by the developer, or where not all authors have a financial interest in the software. We recognize the potential limitations of this; however, we believe the user base who regularly applies these tools would be attracted to manuscript co-authorship with the developers, thus bringing the user community’s specific perspective to the tutorial text.

Finally, regardless of its potential shortcomings, it seems to us that “peer review” will continue to be the method of choice for propagation of scientific information on new discoveries and novel approaches to technical problems, and our journal of course wholeheartedly supports this notion. Software tutorials are not different in this respect and will be handled accordingly.

SUMMARY

Our view is that peer-reviewed software tutorials to be published in CPT:PSP should include: a description of one or more scientific motivating problems that will be used to
structure the tutorial flow; instructions as to how to use the software to solve this or similar modeling analysis problems; scripts or other computer code or pseudocode, as appropriate, to enable a user to reproduce the tutorial sequence; and an inclusive author list, ideally comprising users without monetary interests in the software, in addition to the software developers.

A line attributed to the mathematician Paul Halmos states that “the best way to learn is to do; the worst way to teach is to talk.” After committing some of our thoughts to paper, we now prefer to let our submissions do the talking. Accordingly, we hope that the software tutorials we will host on the pages of CPT:PSP will contain relevant, useful, reproducible, durable, and practical examples to effectively assist our readers in learning, and in turn sharing and thus growing, our craft.

P Vicini1, LE Friberg2, PH van der Graaf3 and A Rostami-Hodjegan4

1Pharmacokinetics, Dynamics and Metabolism—New Biological Entities (PDM-NBE), Pfizer Worldwide Research and Development, San Diego, California, USA; 2Department of Pharmaceutical Biosciences, Uppsala University, Uppsala, Sweden; 3Leiden Academic Centre for Drug Research, Leiden University, Leiden, The Netherlands; 4School of Pharmacy and Pharmaceutical Sciences, Faculty of Medical and Human Sciences, University of Manchester, Manchester, UK.

Correspondence: P Vicini (paolo.vicini@pfizer.com)

Acknowledgments. P.V. receives royalties from the University of Washington, the licensor of the SAAM II software. A.R.-H. is an employee of the University of Manchester and part-time secondee to Simcyp Limited (a Certara company). Simcyp’s research is funded by a consortium of pharmaceutical companies.

1. Friberg, L.E. Tutorials on the foundations of pharmacometrics and systems pharmacology. CPT Pharmacometrics Syst. Pharmacol. 2, e53 (2013).
2. Keizer, R.J., Karlsson, M.O. & Hooker, A. Modeling and Simulation Workbench for NONMEM: tutorial on Pirana, PaNi, and Xpose. CPT Pharmacometrics Syst. Pharmacol. 2, e50 (2013).
3. Guiastrennec, B., Woltenberg, L., Forrest, A. & Al-Oudha, S. AMGET, an R-Based Postprocessing Tool for ADAPT 5. CPT Pharmacometrics Syst. Pharmacol. 2, e61 (2013).
4. van der Graaf, P.H. CPT: pharmacometrics and systems pharmacology. CPT Pharmacometrics Syst. Pharmacol. 1, e8 (2012).
5. Ptič, A & Lapp, H. The PLOS computational biology software section. PLoS Comput. Biol. 8, e1002799 (2012).
6. Harnisch, L., Matthews, I., Chard, J. & Karlsson, M.O. Drug and disease model resources: a consortium to create standards and tools to enhance model-based drug development. CPT Pharmacometrics Syst. Pharmacol. 2, e34 (2013).
7. Li, C., Courtot, M., Le Novère, N. & Laibe, C. BioModels.net Web Services, a free and integrated toolkit for computational modelling software. Brief Bioinformatics 11, 270–277 (2010).