How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox

Francesca Pianosi[1,2,*], Fanny Sarrazin[1,3], Thorsten Wagener[1,2]

[1]Department of Civil Engineering, University of Bristol, UK
[2]Cabot Institute, University of Bristol, UK
[3]Department of Computational Hydrosystems, UFZ-Helmholtz-Zentrum für Umweltforschung, Leipzig Germany
*Corresponding author: francesca.pianosi@bristol.ac.uk

Highlights

- Sharing open-access software has become common practice among researchers, but is rarely followed up with an analysis of its adoption success
- We were able to survey our toolbox users and found that workflows and commented code were effective in enabling users to adopt and tailor methods
- We also realized that effective uptake requires not only guidance for enabling users to produce results but also provision of guidance for their interpretation

Keywords

Research software; Open-source software; Reproducibility; Workflows; Software documentation

Abstract

Open-source research software is an important element of open science. While the number of software packages made available by researchers is increasing, there has been little analysis about their subsequent uptake. We collect basic information about prospective users when sharing our open-source sensitivity analysis toolbox. This enabled us to carry out a user survey to assess adoption success – beyond simply counting download numbers. Survey results confirm the key role of extensive documentation to ensure adoption, to enhance learning and to enable research implementation. We found that workflows are an effective tool to guide users to tailor methods to their problems. However, workflows also need to include guidance for interpretation of results, otherwise sophisticated functionalities are overlooked as their value is unclear. Developing effective documentation requires significant time investment but is essential if the ultimate aim of open research software is to promote the adoption of scientific methodologies and best practices.

1. Introduction

Sharing free, open-source research software is becoming increasingly common in the environmental modelling community. By 'research' software we refer here to software that is developed by researchers as part (but not as the primary focus) of their research activity – in contrast to 'professional' software produced by software engineering companies. Research software is now often distributed openly along with other key research outputs such as datasets, scientific papers or case study applications.
For example, limiting ourselves to the field of Sensitivity Analysis (Saltelli et al, 2008; Pianosi et al., 2016) to which the SAFE toolbox investigated in this paper contributes,

research software that has been made available over the years includes: the updated version of the Simlab framework (Saltelli et al., 2008; JRC, 2015), the MCAT toolbox (Wagener and Kollat, 2007), the GUI-HDMR software tool (Ziehn and Tomlin, 2009), a new version (Poeter et al., 2014) of the UCODE for model calibration and local sensitivity analysis (Poeter and Hill, 1999), a new release of the R Sensitivity package (Iooss et al., 2018), the Python SALib library (Herman and Usher, 2017), and the VARS-TOOL in Matlab/C++ (Razavi et al., 2019).

Motivations for freely sharing open-source software are manifold and have been extensively investigated. Von Krogh et al. (2012) reviewed the ethical considerations that historically inspired the Free Software movement ("*running software that a user cannot inspect, modify and share is considered immoral*"), as well as the intrinsic and extrinsic motivations that have driven developers thus far. Intrinsic motivations (i.e. stemming from a pursuit of internal satisfaction rather than an external reward) include the fun of developing software, a feeling of reciprocity (wherein one helps others because of having been helped or expecting to be helped) and the gratification of recognition by other community members (the "egoboo" drive; Raymond (1999)). We would argue that all these intrinsic motivations apply to the case of research software developers. An additional motivation can be the belief this activity serves science by (1) increasing the uptake of our models and methods through the availability of the software; and (2) by increasing the transparency and reproducibility of our own analyses if the software is open-source (Stodden et al., 2016; Hutton et al., 2016; Slater et al., 2019). External motivation can also be important, and many have advocated for more explicitly rewarding research software development in academic career progression. For example, new journals have been created for publishing research software in the form of short, peer-reviewed and citable papers (examples are *SoftwareX* and the *Journal of Open Source Software*), and specific funding opportunities and research quality metrics have been advocated (Crouch et al., 2013).

We were motivated by all the above reasons when we released our open-source "Sensitivity Analysis For Everybody" (SAFE) toolbox (Pianosi et al, 2015). We particularly had the ambition to facilitate the use of Global Sensitivity Analysis (GSA) by environmental modellers and to promote the uptake of what we considered best practices in GSA application – reviewed in a companion paper (Pianosi et al. 2016). These goals informed our design philosophy, which was characterised by (Pianosi et al, 2015): (1) modularity of the code - in order to facilitate multi-method analyses; (2) high density of comments – to facilitate understanding of the working principles of each method; (3) minimal dependence on specific Matlab toolboxes/versions – to slow down obsolescence; (4) availability of robustness and convergence metrics – to enable rigorous analysis; (5) availability of visualization functions – to support interpretation and communication of results; (6) availability of workflows - to help users get started and to nudge them to follow best practices (enabled by point 1, 4 and 5).

The use of workflows, in place of a user manual, was one of the specific features of SAFE. In general, a workflow is a series of connected steps employed to achieve an overall goal (Duffy et al., 2012). In computational methods, workflows provide the "information that explains what raw data and intermediate results are input to which computations" (Stodden et al., 2016). In the SAFE toolbox, a workflow is an executable script that shows, through a practical example, how the toolbox functions can be put together to perform a sensitivity analysis (see an example in Figure 1). Users can utilize workflows as tutorials to learn how to apply a specific method implemented in SAFE. Or they can use workflows as a starting point to tailor the script for their own application, given that many steps in GSA are common across problems and applications. Encouraging users to produce and possibly share workflows is also an effective way to promote transparency and reproducibility of the analyses. As such, workflows can be seen as equivalent to the “modelling protocols” proposed by Ceola et al. (2015) as a key mechanism to ensure comparability and reproducibility of model comparison studies.

Since 2015, we have distributed SAFE through a dedicated website (safetoolbox.info) while simultaneously collecting basic information about prospective users who submitted a download request (such as their affiliation, area of expertise, etc.). In 2017, almost 3 years from the first software release and having received about 1000 download requests, we carried out a survey of SAFE users to evaluate its level of adoption and the success of the design choices discussed above. In this paper, we present the survey results to reflect on the efficacy of our approach to open-source software design and distribution, and we draw general conclusions regarding the value and effectiveness of different types of documentation, especially workflows.

The dataset of our survey respondents is quite unique. Indeed, many repositories used to share research software (such as GitHub or Matlab central) do not collect either contact details nor basic information of those who download the software. This makes it very difficult to return to potential users for a survey, or to verify whether the sample of respondents to a generic survey is representative of the overall population of potential software users (given that the characteristics of that population would be unknown) thus undermining the statistical significance of the survey results. Although limited in scope, we thus think our results are interesting and provide some insights of general interest to help other environmental modellers and software developers to improve the quality and efficacy of their research software projects. We also hope that they can inspire others to carry out similar surveys that could help tailor their efforts for software development, distribution and uptake.

```
% Number of uncertain parameters subject to SA:
M    = 5 ;
% Parameter ranges (from literature):
xmin = [   0 0 0 0    0.1 ];
xmax = [400 2 1 0.1 1    ];
% Parameter distributions:
DistrFun  = 'unif'  ;
DistrPar = cell(M,1);
for i=1:M; DistrPar{i} = [ xmin(i) xmax(i) ] ; end
% Name of parameters (will be used to costumize plots):
X_labels = {'Sm','beta','alfa','Rs','Rf'} ;

% Define output:
myfun = 'hymod_nse' ;

% Step 3 (sample inputs space)

r = 100 ; % Number of Elementary Effects
% [notice that the final number of model evaluations will be equal to
% r*(M+1)]

% option 1: use the sampling method originally proposed by Morris (1991):
% L = 6  ; % number of levels in the uniform grid
% design_type  = 'trajectory'; % (note used here but required later)
% X = Morris_sampling(r,xmin,xmax,L); % (r*(M+1),M)

% option 2: Latin Hypercube sampling strategy
SampStrategy = 'lhs' ; % Latin Hypercube
design_type = 'radial';
% other options for design type:
%design_type  = 'trajectory';
X = OAT_sampling(r,M,DistrFun,DistrPar,SampStrategy,design_type);

% Step 4 (run the model)
Y = model_evaluation(myfun,X) ; % size (r*(M+1),1)

% Step 5 (Computation of the Elementary effects)

% Compute Elementary Effects:
[ mi, sigma ] = EET_indices(r,xmin,xmax,X,Y,design_type);
```

Figure 1 – Example of workflow included in the SAFE toolbox. The workflow guides users in the application of a particular method, including explanatory comments and references. Users can execute the workflow 'as is' or change some of the input values and evaluate the impact of these choices. They can also use the workflow as a starting point to develop their own application, by changing the lines where their inputs or models differ from the example shown in the workflow.

2. Survey design

We divided the survey questionnaire into 3 main parts (the full survey – with responses - is included in the Supplementary Material):

Part 1 - general information about the respondent, including their expertise in GSA and the extent to which they used SAFE (Q1-6).
Part 2 - specific information on the ways they used SAFE (Q7-14) aimed at testing whether our design choices reached their intended goals.
Part 3 – some final questions about possible future directions for SAFE development (Q15-18).
We used closed-answer questions in order to make the survey easier to complete (we aimed at a response time of about 10 minutes) and to analyse. We included a N/A ("Not Applicable") answer option when needed in order to ensure that also respondents who requested SAFE but did not actually use it were able to complete the survey.
We used the Online surveys (formerly BOS) platform (www.onlinesurveys.ac.uk) to distribute the survey to the 1000 researchers who had requested SAFE at the time (November 2017). We kept the survey open for 10 days and sent two subsequent reminders as the deadline was approaching. One useful feature of the Online surveys' platform is that it can be set to send reminders only to those contacts that have not responded yet.

3. Survey results

In this section, we report some key analyses of the survey responses. Given that, as expected, only a fraction of survey recipients actually completed it, we first compare some basic characteristics of the respondents' sample and the surveyed population (Sec. 3.1), such as their area of study/research and job title. The objective here is to establish whether the sample of respondents provides a reasonable representation of the overall population of SAFE users. We then move on to analyse the survey responses (Sec 3.2), focusing on key points that may be of general interests to research software developers (besides the GSA/SAFE user community). For the interested reader, we report the full extent of the survey responses in the Supplementary Material.

3.1 Analysis of respondents

We received n=195 responses from our surveyed population of N=1000 people who requested SAFE over the period 2015-2017. This corresponds to a response rate of almost 20% and a margin of error (at 95% confidence level) of:

margin of error (at 95% confidence level) = $\sqrt{\frac{N-n}{N-1}}\frac{0.98}{\sqrt{n}}100 = 6.3\%$

(Isserlis, 1918; Sharon, 1999). Figure 2 analyses the basic characteristics of our surveyed population (left) and of the survey respondents (right). The top panels show that the areas of study/research are well represented in the respondents' sample (i.e. their relative extent is similar in the two populations). The bottom panels instead show some differences in terms of roles, as Bachelor/Master students are underrepresented in the respondents' sample, and lecturers/professors are slightly overrepresented. We assume this may be due to the fact that Bachelor/Master students are more likely to disengage from the research area after completing their student projects; some of them may have also not received the survey at all if they lost access to the University email account after graduation.

Despite these small differences, we believe the respondents sample to be an acceptable representation of the target population.

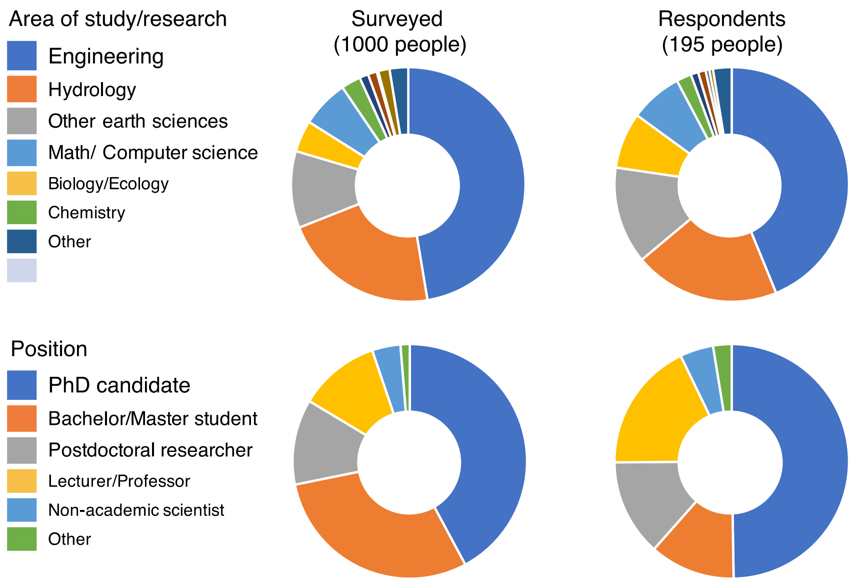


Figure 2 – Area of study/research (top) and role (bottom) of the 1000 people we surveyed as they requested SAFE between 2015 and 2017 (left) and of the 195 people who completed our survey (right).

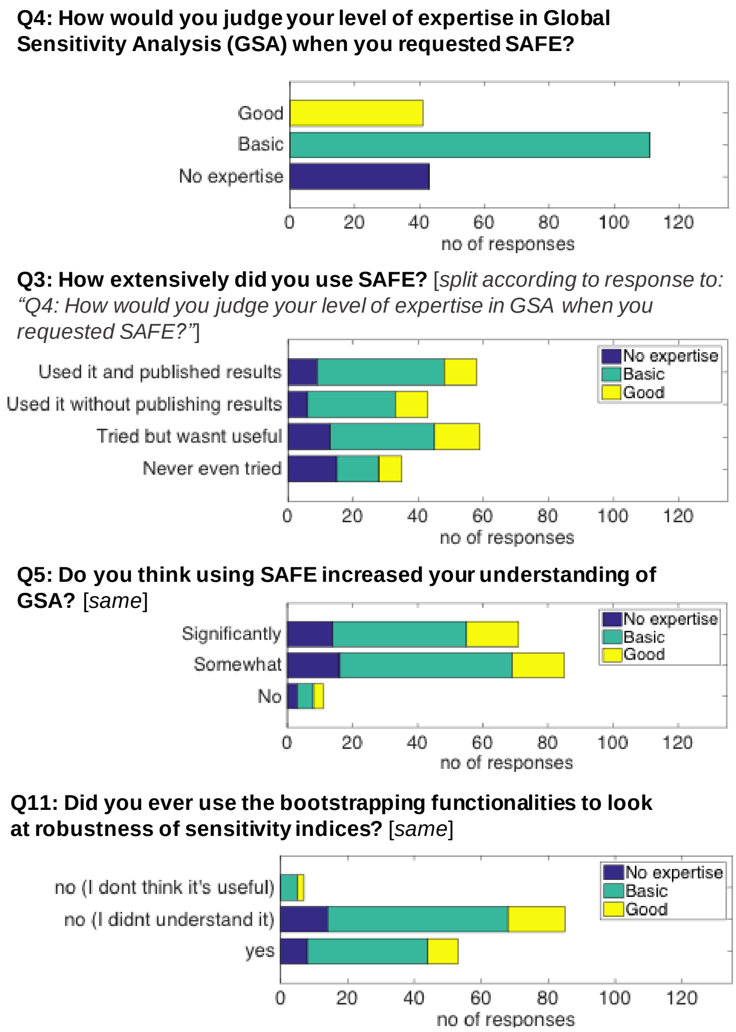
3.2 Analysis of responses

Figure 3, 4 and 5 show a selection of the survey answers and the key points we can draw from them. Beyond specific feedback on SAFE that will be useful for us to improve the toolbox, we believe that the following points of general interest arise.

The availability of open-source software is attractive to users with diverse levels of expertise (among our respondents, 22.1% said they had “no expertise at all” of GSA when they requested SAFE, 56.9% had “basic” expertise, and 21% had “good/advanced” expertise – see Q4 in the Supplementary Material and the top panel of Figure 3). This is in line with our expectations and design choices, which were meant to make the toolbox useful for both experienced and novice users (Pianosi et al., 2015).

Roughly half of the people who requested the software did actually use it. About a fifth (17.8%) instead did not even try it and a third (30.3%) tried it but did not find it useful for their research (question Q3 and middle panel in Fig. 3). This result suggests that, somehow expectedly, the number of download requests of a free software may significantly overestimate the number of actual users.

On the other hand, the respondents who did use the toolbox largely benefitted from it, both through producing useful results (for 29.7% of all respondents up to the point of including them into a publication; Q3) and through improving their understanding of the underpinning GSA methods (43.6% said their understanding increased “somewhat” and 36.4% said it increased “significantly”; Q5 in Fig. 3). This result confirms that sharing open-source software is an effective way to also promote the understanding and uptake of methodologies by a wider range of researchers. Interestingly, the increased understanding seems to be equally perceived by users who started with no expertise in GSA as well as by those who already had a good initial level of understanding.



Key points

SAFE was requested by researchers with a very diverse level of expertise in GSA, from no expertise (coloured in blue in all bar plots) to good/advanced (yellow).

Of those who requested a copy of SAFE, about half (52%) actually used it. Interestingly, about a third of the respondents not only used it but also got to publish results.

Additionally, slightly more than a third said using SAFE increased their understanding of GSA significantly. This seems to apply equally across the (declared) level of expertise in GSA at the time of requesting SAFE.

On a less positive end, almost half of respondents said they did not use the bootstrapping functionalities available in SAFE, despite our efforts to highlight their value for a rigorous GSA application.

Figure 3 – Analysis of the responses to selected questions of the survey about the use and usefulness of the SAFE toolbox and some of its advanced functionalities such as bootstrapping (note: plots of responses to Q5 and Q11 do not include the "N/A" answers by respondents who did not actually use SAFE; full responses to all questions are given in the Supplementary Material).

Some users exploited the functionalities of SAFE in line with our suggested best practice, for example by applying multiple GSA methods (29.2%; Q9 in the Supplementary Material) and by complementing the analyses with visualisation functions (42.6%; Q10). However, more sophisticated functionalities such as bootstrapping, which we consider essential to ensure the trustworthiness of conclusions drawn from GSA results (Sarrazin et al., 2016), was not picked up as much as we hoped. Surprisingly to us, 43.6% of respondents declared they did not use it because they did not understand it, and 3.6% declared they do not think it is useful at all (Q11 and bottom panel in Figure 3). This result suggests that, unless a specific effort is made in highlighting the value of more sophisticated functionalities, many users are likely to overlook them.

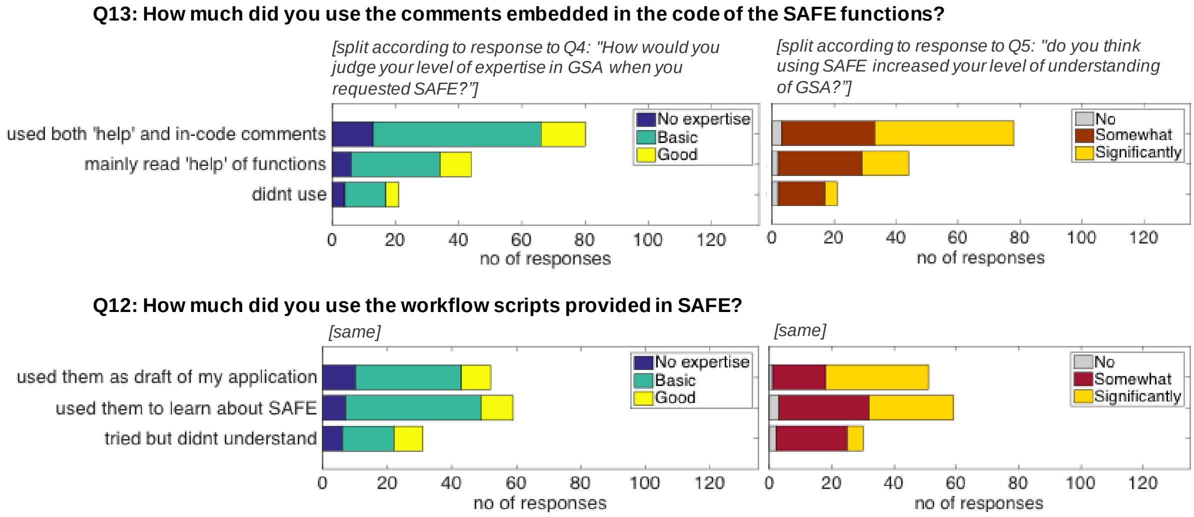


Figure 4 – Analysis of the responses to selected questions of the survey about the effectiveness of the SAFE toolbox documentation (note: plots do not include the "N/A" answers by respondents who did not actually use SAFE; full responses to all questions are given in the Supplementary Material).

Most users read and appreciate comments, both in the form of function help (22.6%) and in-code comments within functions (41%) (Q13 and Figure 4). Again, the repartition of responses according to level of initial expertise is quite uniform (top left panel in Figure 4), which suggests that users with different backgrounds found the documentation equally accessible. However, the top right panel in Figure 4 shows that the users who mostly engaged with the in-code documentation are more likely to perceive a significant increase in their understanding of GSA. This result suggests that sharing well commented open-source software is an effective mechanism to improve understanding of methodologies.

Workflow scripts also proved useful (Q12 and bottom panel of Figure 4): a third of respondents used them to learn about specific GSA methods and about a fourth (26.7%) employed them as an initial draft for their own workflow. Similar to the in-code comments, we see here as well that the users who engaged most with the workflows also found their general learning of GSA improved most significantly.

When asked about possible future developments, many respondents suggested workflows closer to their application area (45.1%) (see Q15 in the Supplementary Material and Figure 5), and the majority said they would also want a user manual (70.3% of respondents said it would be "very useful"). In our approach to developing documentation, we assumed that the scientific background knowledge provided by a manual could be conveyed equally effectively by peer-reviewed papers that we published along with the SAFE toolbox. For example, our literature review paper (Pianosi et al., 2016) includes an extensive description of the available methods in

SAFE as well as the key steps in the set-up of GSA, which are mirrored one-to-one in the SAFE workflows. However, we acknowledge that we must have failed in clearly communicating the availability of such scientific documentation. Indeed the literature review and other scientific papers (such as Sarrazin et al (2016), which provides the background to the bootstrapping functionalities in SAFE) are linked in the FAQ page of the SAFE Toolbox website (www.safetoolbox.info/faq/), which 31.3% of our respondents said they did not realize was available (see Q14 in the Supplementary Material). In order to better highlight these resources we have now modified the email message by which the toolbox is sent to users.

Lastly, in responding to Q14 and/or to the final open-ended question, a good number of users said that they would want a version of SAFE in R or Python. This gave us motivation to accelerate our plans to develop such versions, which have now been made available through the same SAFE website.

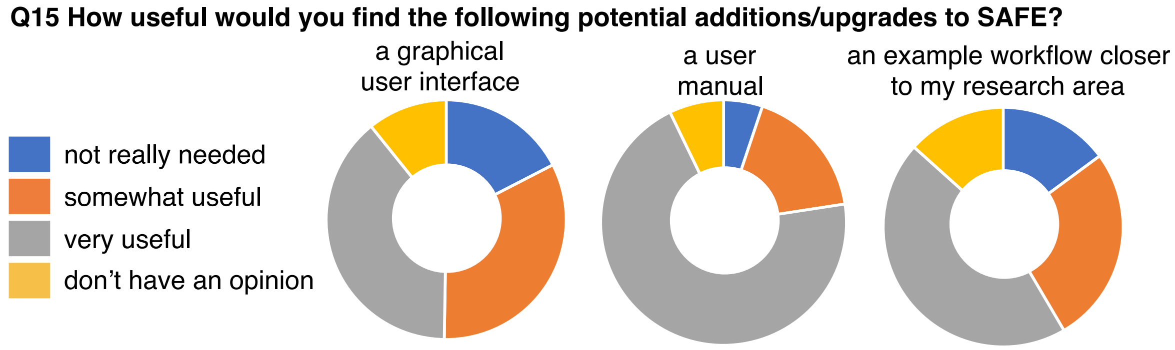


Figure 5 – Analysis of the responses to selected questions of the survey (full responses in the Supplementary Material) about possible additions to the SAFE toolbox.

4 Outlook

In 2015 we started sharing an open-source toolbox for Global Sensitivity Analysis (GSA). We chose to distribute the toolbox through a bespoke website with an electronic registration form, which enabled us to collect some basic information from everybody downloading the software. This enabled us to perform a survey that proved very useful for us to measure the actual uptake of the software and the effectiveness of some of our design choices, as well as to inform upgrades to the toolbox. On the other hand, we are aware that this choice may be frowned upon by some potential users, and it reduces the interaction with the user base and the integration of their contributions – which would both be easier if the code was hosted on a public platform such as GitHub. Given the relatively specialised nature of the toolbox, and hence the relatively small size of its user community, it is still possible for us to manage interactions via email, and we have occasionally uploaded add-ons contributed by users on the website FAQ page[1]. While we recognise this approach may become

[1] Some figures: we had received 1000 download requests at the time of the survey (November 2017) and about 2200 at the time of writing this manuscript (September 2019). We currently receive about one email a month with request for additional clarification or comments from users. We know some users have tailored the toolbox functions to their specific needs (see for example the comments to our survey's open-ended question in the Supplementary material) but they have rarely shared this back

cumbersome when dealing with larger communities or frequent software updates, thus far we have found that the advantages, such as enabling user surveys, outweigh its limitations.

We decided to share our toolbox mainly as a way to increase the impact and utilization of GSA methods. The survey results presented in this paper seem to confirm that software availability has helped other researchers in their applications of GSA - in about 30% of cases all the way to the point of publishing the results. Importantly, the majority of users also declared that using the toolbox increased their understanding of the methods it makes available. We believe that the key to this success lies in the extensive documentation we developed (in the form of function helps, in-code comments, workflows, and a website with links to application examples and Frequently Asked Questions). The survey results further confirm that documentation is essential to both enable the use of the software as well as to realise its potential for increasing the understanding of the underpinning methods.

However, developing a extensive documentation requires time and resources that go well beyond what is needed for the development of the code itself. Such effort was acceptable within the context of the research project in which SAFE was first developed (the CREDIBLE project; NE/J017450/1). In fact, developing software tools was regarded as one of the ways to achieve the broader project goal of improving the consideration of model uncertainty in natural hazard assessment. In other cases, researchers may share their code for different aims. For example, some may ‘just’ want to avoid duplication of efforts and save other researchers’ the time needed to carry out similar developments. In this case, it may be unreasonable to expect researchers to also develop thorough documentation when they are already sharing their work for free. The point we would like to make here is that setting a clear goal for distributing open-access research software determines the amount of effort put into developing the documentation that is needed to reach that goal.

A key lesson learnt from the survey is that, if software is meant to facilitate the uptake of a methodology and its appropriate application, users need to be supported not only in generating results but also in interpreting them. Such a need has been openly recognised by our survey respondents (for example, in response to the final open-ended question, one user said: “*A user guide featuring basic approaches to interpreting results could be very useful*”) and is implicitly suggested by the limited uptake of those functionalities, such as bootstrapping, whose meaning is not immediately self-evident. So, documentation should not be limited to enabling users to produce results but should also provide guidance on interpreting them and for understanding their implications. Users will not adopt methods if they do not understand the value of the information they provide, and they will fall back on the simplest and easiest to understand software functionalities. The survey results also confirm that workflows are an effective tool for knowledge transfer. Though, in order to achieve our objectives fully, workflows should include both guidance on how to produce results as well as how to interpret them. We have hence developed additional documentation and workflows specifically focused on analyzing and interpreting the

with us. So far, we have updated the Matlab toolbox once, and sent the new release to users by email. We recently added the option to download an R and a Python version of the toolbox and we are planning another update of the Matlab version.

meaning and implications of key set-up choices in GSA application (Wagener and Pianosi, 2019; Noacco et al., 2019).

An exciting opportunity for the implementation of user-friendly workflows comes in the form of interactive notebooks that are becoming increasingly easy to develop, thanks to new tools such as the R Shiny package (https://shiny.rstudio.com), the Wolfram Notebook Interface for Mathematica (http://www.wolfram.com/mathematica/) and Jupiter notebooks for Python (https://jupyter.org). We believe that these new packages offer an unprecedented opportunity for developing interactive workflows that are extremely effective in supporting users to explore methods, choices and results in general. Besides supporting knowledge transfer and training, workflows are also highly valuable for increasing the transparency and reproducibility of individual applications (Hutton et al., 2016). Shared workflows (often connected to published journal papers as supplemental material) enable other users to reproduce previous analyses and provide a starting point for users to develop their own applications – thus directly benefiting from previous software/method tailoring. Ultimately, shared workflows provide an agile mechanism to increase the transparency and reproducibility of computational research.

Acknowledgement

The SAFE toolbox was initially supported by the UK Natural Environment Research Council (NERC) through the Consortium on Risk in the Environment: Diagnostics, Integration, Benchmarking, Learning and Elicitation (CREDIBLE) [NE/J017450/1]. F. Pianosi is currently supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through a “Living with Environmental Uncertainty” fellowship [EP/R007330/1]. The authors wish to thank three anonymous reviewers for their comments that helped improving the discussion of the survey results. The authors are also grateful to the many users of the SAFE toolbox who over the years provided their feedbacks and comments through individual contacts and by taking part in the survey presented in this paper.

References

Ceola, S., Arheimer, B., Baratti, E., Blöschl, G., Capell, R., Castellarin, A., Freer, J., Han, D., Hrachowitz, M., Hundecha, Y., Hutton, C., Lindström, G., Montanari, A., Nijzink, R., Parajka, J., Toth, E., Viglione, A., and Wagener, T. (2015): Virtual laboratories: new opportunities for collaborative water science, Hydrol. Earth Syst. Sci., 19, 2101-2117.

Crouch, S., Chue Hong, N., Hettrick, S., Jackson, M., Pawlik, A., Sufi, S., Carr, L., De Roure, D., Goble, C., Parsons, M. (2013), The Software Sustainability Institute: Changing Research Software Attitudes and Practices, Computing in Science & Engineering, 15(6), 74-80.

Duffy, C. J., Gil, Y., Deelman, E., Marru, S., Pierce, M., Demir, I., & Wiener, G. (2012). Designing a road map for geoscience workflows. Eos, 93(24), 225-226.

Herman, J. and W. Usher (2017), SALib: An open-source Python library for Sensitivity Analysis, The Journal of Open Source Software, 2(9).

Hutton, C., T. Wagener, J. Freer, D. Han, C. Duffy, and B. Arheimer (2016). Most computational hydrology is not reproducible, so is it really science? Water Resour. Res., 52, 7548-7555.

Iooss, B., Janon, A. and Pujol, G. (2018), Sensitivity: Global Sensitivity Analysis of Model Outputs, https://CRAN.R-project.org/package=sensitivity (last visited: 18 April 2019)

Isserlis, L. (1918), On the value of a mean as calculated from a sample. Journal of the Royal Statistical Society, 81(1): 75-81.

Joint Research Center (2015), SIMLAB and other software, https://ec.europa.eu/jrc/en/samo/simlab (last visited: 18 April 2019)

Krogh, G. V., Haefliger, S., Spaeth, S. and Wallin, M. W. (2012). Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development. MIS Quarterly, 36(2), pp. 649-676.

Lohr, S. L. (1999), Sampling: Design and Analysis. Pacific Grove, California: Duxbury Press.

Noacco V., Sarrazin, F., Pianosi, F., Wagener, T. (2019), Matlab/R workflows to assess critical choices in Global Sensitivity Analysis using the SAFE toolbox, MethodsX, 6, 2258-2280.

Pianosi, F., F. Sarrazin, and T. Wagener (2015), A Matlab toolbox for global sensitivity analysis, Environmental Modeling & Software, 70, 80-85.

Pianosi F., Beven, K., Freer, J., Hall, J.W., Rougier, J., Stephenson, D.B. and Wagener, T. 2016. Sensitivity analysis of environmental models: A systematic review with practical workflow. Environmental Modeling & Software, 79, 214-232.

Poeter, E.P. and Hill, M.C. (1999), UCODE, a computer code for universal inverse modeling, Computers & Geosciences, 25(4), 457-462.

Poeter, E.P., Hill, M., Lu, D., Tiedeman, C., Mehl, S. (2014) UCODE_2014, with New Capabilities To Define Parameters Unique to Predictions, Calculate Weights Using Simulated Values, Estimate Parameters with SVD, Evaluate Uncertainty with MCMC, and More. GWMI 2014-02. Integrated Groundwater Modeling Center

Raymond, E. 1999. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, revised ed. O'Reilly & Associates, Boston, MA.

Razavi, A., Sheikholeslami, R., Gupta, H.V., Haghnegahdar, A. (2019), VARS-TOOL: A toolbox for comprehensive, efficient, and robust sensitivity and uncertainty analysis, Environmental Modelling & Software, 112, 95-107.

Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, (2008), Global Sensitivity Analysis. The Primer. Wiley.

Sarrazin, F., Pianosi, F. and Wagener, T. 2016. Global Sensitivity Analysis of environmental models: Convergence and validation. Environmental Modeling & Software, 79, 135-152.

Slater, L.J., Thirel, G., Harrigan, S., Delaigue, O., Hurley, A., Khouakhi, A., Prodoscimi, I., Vitolo, C., Smith, K. (2019), Using R in hydrology: a review of recent developments and future directions, Hydrology and Earth System Science Discussion, in review, 2019.

Stodden, V., McNutt, M., Bailey, D. H., Deelman, E., Gil, Y., Hanson, B., Heroux, M.A., Ioannidis, J.P.A. and Taufer, M. (2016), Enhancing reproducibility for computational methods, Science, 354(6317), 1240-1241.

Wagener, T. and Kollat, J. (2007). Numerical and visual evaluation of hydrological and environmental models using the Monte Carlo analysis toolbox. Environmental Modelling & Software, 22(7), 1021-1033.

Wagener, T. and Pianosi, F. (2019). What has Global Sensitivity Analysis ever done for us? A systematic review to support scientific advancement and to inform policy-making in earth system modelling. Earth-Science Reviews, 194, 1-18. doi.org/10.1016/j.earscirev.2019.04.006

Ziehn, T., Tomlin, A., 2009. GUI-HDMR - a software tool for global sensitivity analysis of complex models. Environmental Modeling & Software. 24 (7), 775-785.