Reproducibility in the Field: Transparency, Version Control and Collaboration on the Project Panormos Survey

Néhémie Strupler, Toby C. Wilkinson

Original Study

Néhémie Strupler*, Toby C. Wilkinson

Reproducibility in the Field: Transparency, Version Control and Collaboration on the Project Panormos Survey

https://doi.org/10.1515/opar-2017-0019
Received October 19, 2016; accepted July 10, 2017

Abstract: Archaeological fieldwork is rarely considered reproducible in the sense of the ideal scientific method because of its destructive nature. But new digital technology now offers field practitioners a set of tools that can at least increase the transparency of the data-collection process as well as bring other benefits of an Open Science approach to archaeology. This article shares our perspectives, choices and experiences of piloting a set of tools (namely: *ODK, Git, GitLab CE and R*) which can address reproducibility of fieldwork in the form of an intensive survey project in western Turkey, and highlights the potential consequences of Open Science approaches for archaeology as a whole.

Keywords: Open Science, multi-vocality, decentralisation of data, reproducibility, post-field collaboration, transparency, ethics, fieldwork

1 Introduction: Open Science and Archaeology

> "Open science is the idea that scientific knowledge of all kinds should be openly shared as early as is practical in the discovery process" (Nielsen 2011).

What seemed irrelevant to one generation of scholars is often of paramount importance to the next. Even predicting what data will be of use to other contemporary scholars can be hard. The default academic publishing genres, involving the careful selection and exclusion of data as support for particular arguments or narrative (cf. Bazerman 1983, Knorr & Knorr 1978), while essential for the process of academic dialogue, therefore risks the loss of collected information that, even if they appear to lack important insights to the data creators, may yet serve future research (Nuzzo 2015).

As a science, archaeology relies on empirical research and the production of new data. However, the current practices of result dissemination are often opaque (Morin et al. 2012), the methodologies which translate finds to summaries are frequently difficult to access, and our abilities to re-use data or methods are impeded by the spectre of the final report (Kansa 2012). Though an excavation or a survey cannot be reproduced in the same manner as controlled experiments in the biological or physical sciences, a proportionate degree of reproducibility and transparency still needs to play a part in enriching the process of field data dissemination and opening the possibility of critique through result-checking (Fanelli 2013) and re-use.

***Corresponding author: Néhémie Strupler,** Archéologie et histoire ancienne: Méditerranée-Europe, ARCHIMÈDE (UMR 7044), Université de Strasbourg and Institut für Altorientalische Philologie und Vorderasiatische Altertumskunde, Westfälische Wilhelms Universität Münster. E-mail: nehemie.strupler@etu.unistra.fr
Toby C. Wilkinson, Churchill College/McDonald Institute for Archaeological Research, University of Cambridge

One of the major aims of Open Science, as aptly defined by Michael Nielsen above, is precisely to mitigate these risks by more open publishing practices (analyses, commentaries and data) in order to augment the scientist's traditional, more rhetorically-oriented writing styles (Bartling & Friesike 2014). Archaeologists in particular have much to gain from new models of publication in which data are released on-line and may be updated, corrected, or completed in a more dynamic, interactive and open-ended manner (Heller, The, & Bartling 2014), a style of working called the 'Push and Publish' model by Eric Kansa et al. (2014). Archaeological research already relies heavily on the (re)use of data from previous studies, incorporating and comparing them to new finds. Re-examination of older work may overturn established, strengthen previous or create new paradigms (Latour & Woolgar 1979). Both fieldwork and post-fieldwork analysis are highly relational and intertextual: e.g. dating objects usually entails an explicit link to finds from previous projects, usually published by different archaeologists. At the moment, however, our modes of publication create unnecessary duplication of effort (cf. Fitzpatrick 2011): the closed format of paper publications (or paper-surrogate formats like PDF) makes bringing sources of data together a lengthy process of extraction, re-organisation and re-digitisation. What is missing is a sensible way to create shared *cumulative* datasets or common procedures to link digital datasets for shared advantage (Schloen & Schloen 2014, Vinck & Clivaz 2014).

Despite the potential advantages to archaeological interpretation through re-use and re-flection, there is certainly some reluctance to adopt an Open Science approach, at least where it applies to data and methods rather than just publications (Harley, Acord, Earl-Novell, Lawrence, & King 2010, Kelty 2014). First, just as digital copying has made plagiarism easier, there remains a fear that someone will take advantage of someone else's efforts, i.e. to steal their work, often known as 'scooping' (Masic 2012). In part, this is a result of the low incentives given to data work compared to final reports or syntheses, and the difficulty of tracking the contribution of specific researchers to data production (Wallis, Rolando, & Borgman 2013). Real evidence about the prevalence and impact of scooping remains poor, however.[1] Second, transparency of data and methods is perceived as risky because it has the potential to reveal mistakes. This opens archaeologists – both projects and individuals – to critique and attack from within academia, but also from outside, especially in sensitive political environments. Finally, there are ethical issues relating to protection of heritage from harm, akin to the protection of human subjects and their data in social or medical research, which have not been explored fully (Lagoze, Block, Williams, Abowd, & Vilhuber 2013).

Whether archaeologists like it or not, the tenets of Open Science are likely to become unavoidable in the near future, as funding agencies and governmental bodies demand ever increasing transparency (EU 2016, HEFCE 2015). As such, it is essential that the discipline has a firm evidential base for establishing its position on Open Science (Kansa 2014), identifying good practice and red lines which need to be defended through real case-studies. It was thus with these critical concerns as well as the potential benefits in mind that the Project Panormos team set out to design a fieldwork methodology which placed Open Science at the centre of a modern archaeological intensive survey project.

In this paper, we discuss the implementation of our pilot of an Open Archaeology project, using a concrete set of approaches and software tools to collect, track and disseminate data from the Project Panormos Survey. As will be explained below, our priority was to increase the *reproducibility* and *transparency* of our work through digital data management. Whilst excavation or survey cannot be directly compared to controlled experimental work, analytical reproducibility can be achieved by making explicit and accessible the data collection strategy and the post-field research pipeline. That is, the intermediate work done by the team after the data were collected into the system to create the various outputs, including presentations, statistical analysis, maps and publications. Thus, the aim of the pilot was to facilitate easy data reuse using a relatively simple but transparent infrastructure, and to implement as much of the spirit of Open Science as seemed to us meaningful and practical. Additionally, in part because of our frustration that such data-work normally gains so little wider recognition or academic credit, despite its importance, we also wanted to

1 Open publication simultaneously makes plagiarism easier to identify through detection software and free-text search such as IEEE CrossCheck Portal for manuscripts. Many Open Data advocates argue that open publication may be the best defense against plagiarism (Creative Commons 2013, Journal of Open Archaeology Data 2017).

find ways to track contributions. The results of this pilot and our developing implementation are therefore presented as an opening contribution to what we hope will be an ongoing discussion of the usefulness and limitations of Open Science in archaeological fieldwork practice (Bevan 2012, Lake 2012, Hugget 2015).

2 The Project Panormos Survey: Field Methods and Innovations

The Project Panormos Survey grew out of an earlier three-year rescue excavation at a site near the presumed ancient Ionian harbour town of Panormos (modern Mavişehir, Aydın, Turkey), the port-of-entry for the oracle sanctuary at Didyma (modern Didim), which had revealed a densely-used necropolis dating to the 7th and 6th centuries BC. The project, a collaboration between the local archaeological museum at Milet (Balat) and the German Archaeological Institute (or DAI, Istanbul), led in the field by Dr. Anja Slawisch (DAI), set out from the beginning to maintain a wide research agenda. To address the horizontal extent of the necropolis, to determine its relationship to any potential contemporary settlement in the area, and to place the Panormos necropolis, in 2015 *Project Panormos* expanded its brief to an intensive survey. Methodologically, the aim was to adopt some of the established intensive field-walking techniques which have long been applied in other parts of the Aegean and in the wider Mediterranean (Alcock & Cherry 2004), but still have been used by a very small proportion of survey projects across Turkey and ancient Asia Minor (with some exceptions: Ersoy & Koparal 2008, Ersoy, Tuna, & Koparal 2010, Matthews & Glatz 2009, Düring & Glatz 2016). For the first season (2015), the experiences from which this article is based, it was decided to focus on the region immediately around the archaic necropolis in order to identify the extent of the site and to identify any nearby settlements and landscape usage. Given the relative proximity of a major Bronze Age site of Tavşan Adası, the survey also hoped to identify diachronic landscape use over the long term. A considerable amount of ground was covered in 2015 using a GPS-based grid that allowed relatively regular sampling shapes and hence fast tract recording (Figure 1).

Figure 1. Team entering data after walking a tract in the landscape around Panormos (Photograph by: Toby C. Wilkinson).

Beyond the specific archaeological aims, two interrelated innovations were employed as part of the Open Science-inspired pilot (references and more detailed information describing the installation and interrelationship of the software used can be found in the appendices and the tables). These were instigated and implemented by N. Strupler and T. C. Wilkinson.

1. *Born-digital field data entry system*: The first innovation was the application of in-the-field data entry using the same handheld device, which was also used for GPS navigation. Using the *OpenDataKit* (henceforth *ODK*) software suite, tract data which normally would have been recorded in notebooks or on paper forms was entered immediately by the team leader at the end of each tract.
2. *Distributed version-control data management platform*: The second innovation was the application of a distributed version-control system (*Git*) and a related web-platform (*GitLab CE*) to facilitate the project's data management and the collaboration between team members (whether in the field or beyond). Data and text of all types was structured into purpose-specific *Git* 'repositories'.

Additionally, the version-controlled data-management platform was also used to manage and enhance two diverging but commonly employed practices within survey archaeology, namely prose-oriented reflexive commentary and GIS-oriented mapping of survey results:

1. *Methodological transparency: grey literature, reflexive commentary, multi-vocality and transparency of data collection*: The project's 'daily log' was entered and reviewed into one of the *Git* collaborative repositories, which also held more general information about the project methods, aims and practicalities in the field for the whole team. The long-term aim was also to make this archive open alongside the data as an equivalent to an open lab diary and a re-usable record of the way that the field methods were designed.
2. *Computational reproducibility of survey mapping and analysis*: Rather than simply providing finalised maps as output, we used a scripted approach to GIS (=geographical information systems) by creating procedures which made explicit the process of re-organising data and correcting mistakes, as well as recording the combination with pre-existing spatial information. Code for this analysis was written using *R* and stored in a *Git* repository, with the aim of enabling its evolution to be tracked and for readers to review, reproduce and make their own modifications to the analysis and visualisation.

3 Innovation 1: Born-digital Field Recording: *ODK* and Raw Data

One of the potential epistemological objections to Open Data in archaeology is the idea that archaeology, unlike many 'hard' sciences (e.g. meteorology), rarely has access to the sort of large-scale collection of objective raw measurements (especially through scientific instruments) that warrants digital distribution. With the expansion of geophysical prospection and the oncoming juggernaut of automated morphometrics (e.g. 3D scanning) and easy photogrammetry, one can easily question the scale argument. But even the digitisation of low-level primary observations from the archaeological method can benefit from early digitisation and subsequent release. In the past, most of such observations were collected in excavation or survey log books or, more systematically, in the form of standardised paper forms. With the advent of computing, data from the types of records have been digitised through transcription into digital documents or grid databases and, increasingly, various archaeological projects have experimented with creating digital recording forms directly into the field (e.g. Ellis 2016, Wallrodt 2016). This 'born-digital' approach helps to decrease the distance (and time) between field observations and incorporation of the data into results and dissemination.

Nonetheless, at the time of the project design in spring 2015, we were unaware of any other intensive survey project in Turkey which had implemented a truly 'born-digital' data collection. We selected a free and Open Source software *OpenDataKit* (or *ODK*) as the platform for our pilot. *ODK* is an extremely flexible and easy to implement software suite designed to facilitate digital surveys; originally for social or medical research undertaken in the developing world by individual data collectors on a mobile device (e.g. phone) before being aggregated across an internet connection into a central database for analysis (Anokwa, Hartung, Brunette, Lerer, & Borriello 2009).

For the survey in Panormos, we used two handheld mobile devices (Android-based Garmin Monterra outdoors GPS devices), on which we installed both *ODK Collect* and a number of GIS/mapping apps to help navigation in the field (*AlpineQuest* proved the most useful). Important for us was *ODK*'s ability to work in the field without an internet connection and then to upload the collected data automatically once back at base where WiFi was available. *ODK Collect* primarily serves multiple choice and free-text questions (using text or icons), but can also capture GPS coordinates and photographs with each form. This feature is ideal for linking data together, which is often separated in traditional recording schemes. More details about the specifics of our implementation and form design can be read in the appendix (ODK), but the *ODK* platform proved to be very easy to implement, stable and robust (aside from a documented and solvable glitch with the server setup, which has been fixed in subsequent updates), and enabled access to survey data in digital form even better than we had hoped. Nonetheless, as we expected, human error could not be entirely excluded from the data collection process, and this was clearly evidenced in the process of combining spatial-data (digitised from paper maps) with tract-data (drawn directly from the export from our ODK forms). This will be explained below in greater detail.

In standard fieldwork practice such errors, if recognised, often are corrected at the point of digitisation or at some later stage in publication preparation. Effectively, the existence of mistakes and errors is erased or obscured. In our view, providing only the final corrected versions creates a dangerous fiction: dangerous because it suggests false authority, but also because the corrections themselves may be subject to errors that then become much harder to excise. An Open Science approach effectively demands that data, whether born-digital or not, must have history; such data history can apply to methods and procedures whether digital or not.

4 Innovation 2: Tracking Changes and Contributions, Version Control and Data History with *Git*

Transparency through data history therefore became an essential part of improving the quality and robustness of our pilot and its results. In the past, data history has been preserved through the medium of paper documents (where archives of excavation notebooks or forms are preserved and can be compared to final publications). In a digital age, the intermediate steps can be documented with much greater granularity if data becomes digital at an early stage. This means error corrections or other changes can be tracked more easily, reversed or reflected upon, and usually linked to particular contributors. In the context of computer science, platforms which deal with data history are known as 'version control systems' (or VCS). VCS originated from the need for programmers to trace errors in and keep track of the latest updates to code, especially in collaborative environments. The same principles can apply to different types of data. Version control systems allow the presentation and retrieval of both the most recent version of any particular object and all of its previous incarnations. By way of a concrete analogy, VCS is the equivalent of applying dates to all the drafts of a written paper, keeping them sorted by that date and highlighting the changes made at each stage (and who made the changes).

There are many possible forms that digital version control could take and there are various platforms currently available (an incomplete list would include: *CVS*, *Subversion*, *Git*, *Mercurial*). In recent years, 'distributed VCS' (hereafter DVCS), implemented in frameworks such as *Git* or *Mercurial*, have gained considerable support and application in the field of computer coding. The advantage of DVCS is that it allows the creation and preservation of complex version 'trees' in which multiple authors can contribute to the full history of a project independently. DVCS also includes procedures to manage the merging of conflicting versions, to preserve 'dead-ends' and errors, and simultaneously to maintain a 'head' (i.e. latest version) that can be seen as the endless evolving 'canonical' version.

To manage the pilot survey's data, we decided to use *Git*, a cross-platform, actively developing Free Software DVCS. The software's specifics and the motivations behind its selection are explained in more detail in the Git appendix.

One remarkable observation to be made about a data flow strategy for fieldwork oriented around Open Science policies and a version control platform such as *Git* is the parallel between the task of an archaeologist (who creates a story of the past based on searching for relations between objects/traces in the present) and the task of an Open Science reviewer or a data historian (who creates a story about a project's past based on searching for relations between fragments of data left in the archive, cf. Germán, Adams, & Hassan 2016). The metaphorical link to a process of excavation is even recognised by *Git* developers: one tool is called the `pickaxe interface` and allows users to search for specific strings across the entire *Git* history. *Git* history diagrams are effectively chronological tree diagrams not dissimilar to Harris Matrices. Archaeologists, of all data scientists, therefore should appreciate the importance of data versioning from a practical and theoretical perspective, recognising the insights to future generations that discarded fragments can provide about its producers.

For practical purposes, the full range of features that *Git* makes available to navigate repositories are vastly more sophisticated than the needs of most field archaeologists, and tools to make its use genuinely easy for the average non-coding researcher have yet to appear. Much of the power and control of *Git* to manage repositories can be unlocked with the command-line, since it offers the most control. *Git* also relies to a large degree on UNIX-style user permissions and access controls to manage who can and cannot contribute to or draw from repositories hosted on different machines. For collaborative projects, such access control is essential, but difficult to manage. Even graphical interface (GUI)-based tools available to manage *Git* repositories require a basic knowledge of the principles of *Git* and even if it is possible to avoid the command line for long periods, it may be necessary to return to it for some tasks. Whilst there are many sources of help on the internet to cope with common problems (Chacon & Straub 2014), *Git* still requires IT experience, training and a certain level of confidence. All this may appear to place a high barrier to the take-up of version control into archaeological field practice. With this in mind, we started to look for more user-friendly ways of integrating *Git* and important associated features such as user management into the project's workflow. We settled upon *GitLab*, or rather its free and open source variant, *GitLab CE (Community Edition)*.

4.1 *GitLab CE*: User-Friendly Data Version-Control and Access Management

GitLab CE is a mature web platform which provides a visual web-based interface to track *Git* repositories, as well as a suite of functions needed to manage collaborative digital projects such as team members and access permissions, a communication platform for issues (e.g. bugs or task lists), and systems for automated tasks (i.e. continuous integration and triggers). Like *Git*, it is primarily designed for managing computer code projects, but the principles are surprisingly applicable to more general data-project management.

There are a number of ways that *GitLab* can be run (more details can be found in the GitLab appendix), but for Project Panormos, we chose to set up our own server instead of relying on an external infrastructure. This allowed us considerable scope to experiment with creating, transforming and deleting repositories as our ideas developed, as well as to make mistakes along the way. The *GitLab* CE interface allows the user to navigate versions and edit text-based files directly using the web-interface with individual logins for each team member. Once running, a number of different *Git* repositories were set up using *GitLab CE* to separate different activities within the survey project (see figure 2).

Each project repository was assigned access controls, managed using the *GitLab CE*. For example, only those responsible for digitisation of the tracts or using the resultant GIS files for analysis were given appropriate read or write access to the `survey-spatial` repository. The same was also true of the `survey-data` and `survey-analysis` repositories. Whilst, in retrospect, the level of restriction adopted (repositories only visible to those using them) was probably not necessary within the project, it allowed team members who were new to the system to focus exclusively on those aspects of the project which were relevant to their particular tasks.

Besides management of the *Git* repositories and their histories (including a neat 'merging' procedure to facilitate conflicts between versions), the current *GitLab CE* interface also included an editing facility for plain-text, which proved useful especially for the daily log (see below) and for small corrections to text

or other plain-text data. Our usage of the platform and its functionality for archaeologists and other data scientists would certainly be enhanced by a greater range of in-built web-based editing and comparison ('diff') tools (cf. Smith 2013), e.g. for geographic or tabular data types (e.g. *CSV*, *GeoJSON*). At the time of writing, such facilities exist in *GitHub*, a commercial competitor to *GitLab*, and are in development for *GitLab*.[2]

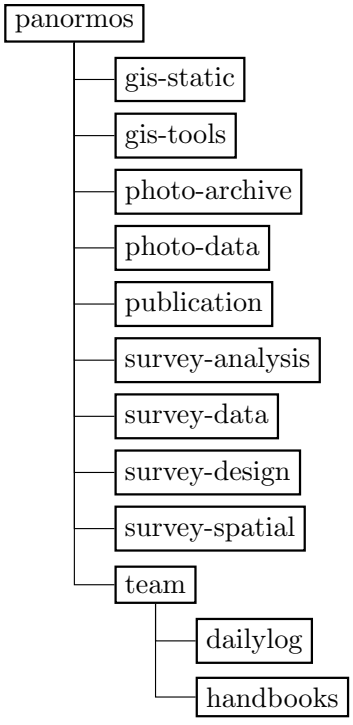

Figure 2. The current structure of the various Project Panormos repositories with an example of the internal structure of the `team` repository.

5 Enhancement 1: Methodological Transparency in the Project Panormos Survey

We realised at an early stage that the same platform (i.e. *GitLab CE* used to manage *Git*-based version-controlled repositories) could be just as useful in the organisation and preservation of more prose-oriented data relating to the project, as the more tabular sorts of information extracted from *ODK*. This included various planning documents we were producing as part of the training of students on the project (project handbooks) and the record of a more narrative sort about the enactment of the project (daily logs).

Much of the 'grey literature' of archaeological fieldwork (e.g. blank and filled forms, fieldwork handbooks, communications between the team) does not make it into publication. Though there are sometimes good reasons for this, being able to accurately and fairly assess the results of a research programme demands that archaeologists know *how* the data was collected as much as *what*. An Open Science approach encourages this kind of methodological transparency.

2 See the discussions on the Gitlab development pages for the latest on this: https://gitlab.com/gitlab-org/gitlab-ce/issues/24287 and https://gitlab.com/gitlab-org/gitlab-ce/merge_requests/10566 (May 18, 2017).

5.1 Project Handbooks: Modulation and Dissemination of 'Untraditional' Output

During the planning stages of the project in spring 2015, the directors of the project created a series of documents outlining the planned strategy of the survey, which included: guides to the survey methodology, use of equipment, first aid advice and instructions on personal conduct, alongside more informal pieces of practical information such as restaurant recommendations or travel information, etc. The aim was to make a transparent record of our methods and policies, as well as their changes through time. Documents were updated as new information became available. Additionally, by making these documents collaborative and dynamic, we also saw that by releasing them we would not only make our own work more transparent, but we would also facilitate the creative re-use of our by-product. Whether this will indeed become the case is currently difficult to tell.

Both fieldwork methodology and archaeological interpretation necessarily involve the accumulation of a pool of knowledge and a set of practices which are based on earlier operations and adapted to new places, but this inheritance process is often implicit. This is what C. Kelty calls *modulation*, the practice of looking at a project, understanding how it works and re-using it for different goals (Kelty 2008). This is exactly the principle used in Free Software, where lines of code are re-employed and modified, small programs integrated within others, all through established licensing and proper attribution to source (Kelty 2005). In the world of research, there is, of course, a built-in resistance to overt *in extenso* modulation of text, since it is seen as a kind of plagiarism, even when sanctioned. This is despite the fact that other forms of covert modulation (particularly re-using data) are widespread. There are many cases where leveraging modulation is self-evidently beneficial, particularly in guides to good practice or form sheet in order to avoid wasted time in needless reinvention. Even if each project has its particular methodology, common basics allow greater cross-comparison, so publication of digital data-entry forms would give a new project a foundational template to work with or allow it to re-use the same templates for multilingual interfaces. At the pinnacle, one can imagine a standard, collaborative and version-controlled *handbook for archaeological practice*, which could be maintained and edited by multiple projects and kept up-to-date with the evolution of the software and methods. Platforms which espouse the Open Science approach can allow the creation of an environment where all in the community are imagined and recognised as potential collaborators (Strupler 2016).

5.2 The Daily Log: Narrative Recording and Multivocality

The project's 'daily log' was an effort to promote multi-vocality (with some of the previous archaeological efforts in this field as a backdrop: e.g. Hodder 1999, 2008) and to pilot the equivalent of open lab notebook or blog within the project. Through a narrative style and through sharing the responsibility of the diary writing (each team member took a turn at writing up the log), the idea was to throw a more colourful and nuanced light on some of the personal dynamics and events which tend not to be recorded in dry archaeological reports but which help contextualise the creation of archaeological data. Of course, with the level of transparency that a version controlled digital log creates, a higher degree of responsibility and care is necessary than might have been typical for traditional paper notebooks, which are rarely made public until considerable time has passed.

On arrival in the field, all team members were given usernames on the *GitLab CE* server and assigned access to the `team` repository, which contained both the project handbooks and blank templates for the daily log (figure 2). Henceforth, team members were allowed to edit the `drafts` branch of the data version tree, writing all the entries using *Markdown* format plain-text. Authors were encouraged to keep entries professional and friendly; commentators (which could be anyone on the team) were likewise encouraged to provide positive feedback and offer editing suggestions once an entry had been made. We used the 'Issues' feature of *GitLab CE* to enable comments. After auditing for errors and other issues, the owner of the `team` repository (i.e. the project directors) approved a 'merge' of the drafts into the `master` (i.e. canonical) branch. This in turn triggered a basic form of 'continuous integration', in which a web-version of both the handbooks and logs (using *Jekyll* static site generator) was automatically generated and uploaded to a web server.

Without exception, all our team members entered their daily log using the *GitLab CE* system, and despite our worries that the interface might be distracting (there are many features not relevant to the task of daily log writing), most found the system straightforward to use and were comfortable with the *Markdown* format of the text entries. In only one recorded case was a long entry lost due to a mistaken web-page refresh.

Disappointingly, there was a fairly low level of commentary on other team member's entries. We suspect the main reason for this was the fact that team members already had multiple opportunities for in-person discussion and interaction, and thus spending time online to communicate with someone with whom you were just about to eat dinner was overly artificial. Also, the 'Issues' feature of *GitLab CE* does not directly parallel the sort of comment features seen in many online blogs, which may have been more intuitive. This suggests that different ways of capturing in-project archaeological dialogues are needed than the default set-up offered by something like *GitLab CE*, which is more commonly employed in desk-based environments with co-workers located at a distance.

6 Enhancement 2: Computational Reproducibility of Survey Mapping and Analysis

Since *Git* and *GitLab CE* were designed for code, it was natural to incorporate the project's computational analysis into the same platform. Computational research is rapidly growing in scale in archaeology, but the degree to which the particular outputs of analysis (whether statistical results, maps or graphs, etc.) are easily reproducible by others has lagged considerably. Making these kinds of visualisation reproducible (as well as the analyses and procedures upon which they are based) is an essential part of an Open Science approach to facilitate testability, falsifiability and replication (or modulation) in other contexts. Instead, producers and consumers sometimes treat computation in archaeology as a kind of black magic, each for different reasons.

Our own application of computational reproducibility for the Panormos Survey derived, in part, from a frustration with which computational approaches are commonly documented in archaeological literature: i.e. brief references to establish (externally-derived) complex procedure (e.g. 'Monte Carlo/ Bayesian analysis are utilised in this paper') with a purely prose explanation. Such papers neither help the expert to reproduce the procedure nor the uninitiated to assess the real worth of the application. Similarly, the procedure to create certain visualisations (maps, graphs) for any particular project are rarely made available (let alone the raw data behind them), meaning that it is relatively hard or time consuming to reproduce or modify. Although, as some have rightly argued, simply mimicking data, script and output is not the same as understanding an analysis (Bissell 2013). There are inherent dangers of potential misuse of complex procedures. But without good exemplars, it is near impossible for those new to a field to learn good practice effectively.

Computational reproducibility has three requirements (cf. Marwick 2016). First, although it may be obvious, the data behind any particular procedure needs to be released openly in a clean, organised and well-documented manner. Ideally this is done in an open and non-proprietary format to avoid rights and access issues. Second, the software used to perform the procedure needs to be Free Software (i.e. ideally in both senses of the word), to allow everyone to obtain the software and rerun the procedure and, if necessary, adapt that software. The same can be said for the operating environment (i.e. operating system and versions) that the software is run: this is being increasingly facilitated by tools such as virtual machines and operating system containers. Finally, interactions between computer and human users need to be recorded in order to reproduce the steps of the analysis. This could involve a list of instructions or, somewhat more tediously, a transcript of mouse clicks or keyboard presses. It is more realistic and effective, however, to record the process of interaction as a script (or code) in a common, again preferably Free/Open Source language, which itself can be distributed and re-run.

For many in archaeology, for whom using GIS to visualise results is essentially a graphical-based point-and-click process, advocating a return to code may seem like a backward step. We understand the arguments for usability, and acknowledge that intermediate tools which can bridge point-and-click with code-based

approaches are desperately required. However, we contend that for serious academic publication, code-based GIS ought to be the default for rigorous production of data processing, analysis and the creation of any kind of resulting visualisation. Scripts can be equally important as part of the data documentation process (Bowers 2011), for example in the case of a `cleaning` script, which simultaneously documents corrections to raw-data, as discussed above, and explains obscure aspects of the data schema or content (cf. Kansa & Whitcher Kansa 2013).

For the purposes of the 2015 pilot, we choose to use the programming language *R* (R Core Team 2017) to create both 'cleaning' and 'visualisation' scripts (for specific details, see appendix R). The motivation behind this selection was partly our familiarity with the language, but also its widespread usage in the context of 'literate programming' (enabled by the `knitr` package in particular), making it easy to embed R code into a prose-oriented document (Xie 2015, 2016). Literate programming is a term, common in the world of *R* and other data science contexts, which refers to the creation of executable documents that include both the code needed to achieve a task computationally and the explanation of the data, procedures and algorithms used: "[I]nstead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do" (Knuth 1984 p. 1).

For the provisional post-fieldwork analyses designed to visualise our results at the end of the season, N. Strupler wrote a number of `cleaning` scripts organised under the `survey-data` repository and *RMarkdown* documents organised under the `survey-analysis` repository. The scripts in the `survey-data` repository process the raw data as exported from ODK, correcting errors and then export to a cleaned `canonical` version. The analysis script, which was written in *RMarkdown*, described the process of import of this canonical data, the transformation of this data into appropriate structural format for analysis, and then its output into a series of visualisations of, for example, pottery density, using customised colour, etc.

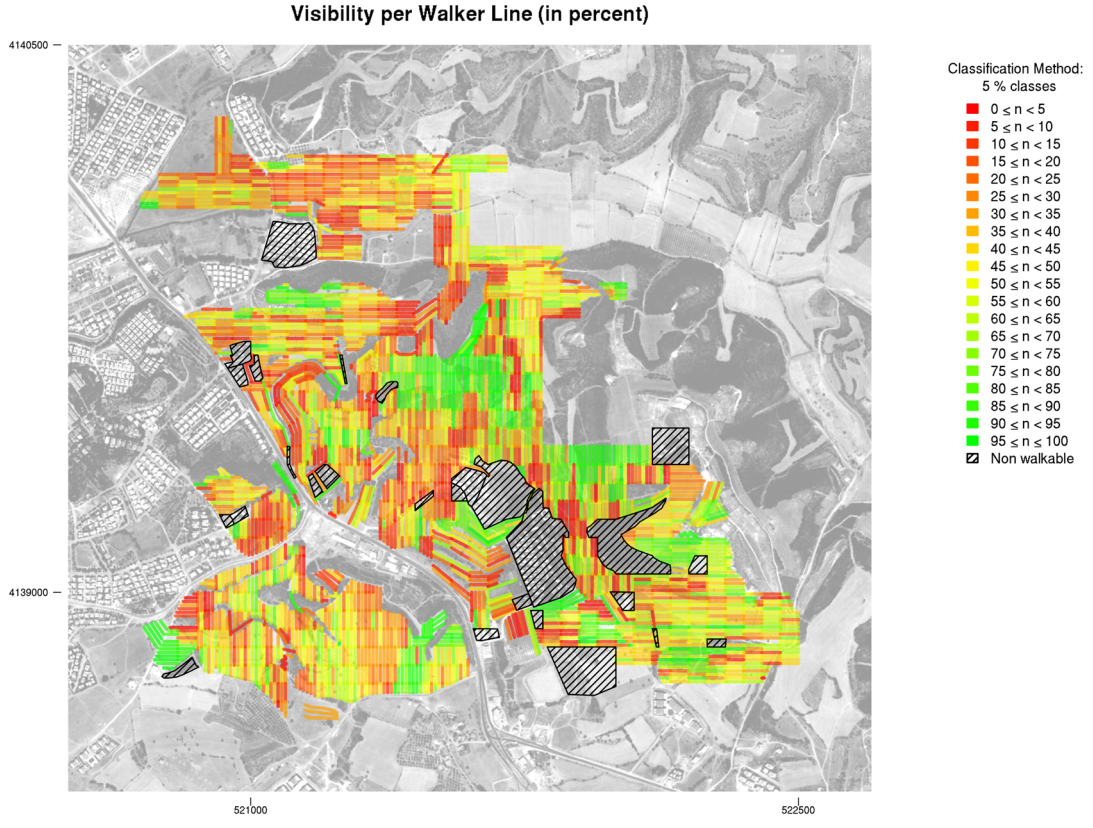

Figure 3. A map plot generated using R, based the 2015 data, showing the ground-surface visibility of each walker. (Base satellite image: WorldView-2, © DigitalGlobe).

Managing this code in a *Git* repository had further benefits: first, the history of how it was coded could be tracked (so that changes which introduced errors might be more easily spotted); second, the code could be immediately shared amongst other team members with appropriate access who could replicate, review the output, or make new modulations.

6.1 Data Flow in the Panormos Survey: Production, Modification and Outputs

Combining these different software platforms allowed us to design a data flow strategy for the whole survey which would take advantage of fast digitisation (i.e. born digital data), version control and granular access to shared data sets. It also established transparent and semi-automated code-oriented cleaning procedures and creation of visualisations which combined the relevant data sources. The diagram shown in figure 4 illustrates an idealised summary of that flow within the project and the relationship between the software and different categories of data. The diagram is organised in broadly chronological stages from top to bottom. The middle section of the diagram indicates the central management role of the VCS (here *Git* and *GitLab CE*), split in sub-projects, i.e. repository (survey, gis-static, survey-analysis, etc.). This schema, which developed organically through the process of project planning and execution, is designed to allow the release of each subproject at a different time point with a specific license when appropriate. Similarly, material used internally by the project, whose copyright is third-party owned, could be kept separate, as for potentially sensitive material (e.g. location data), which could be accessed by request only.

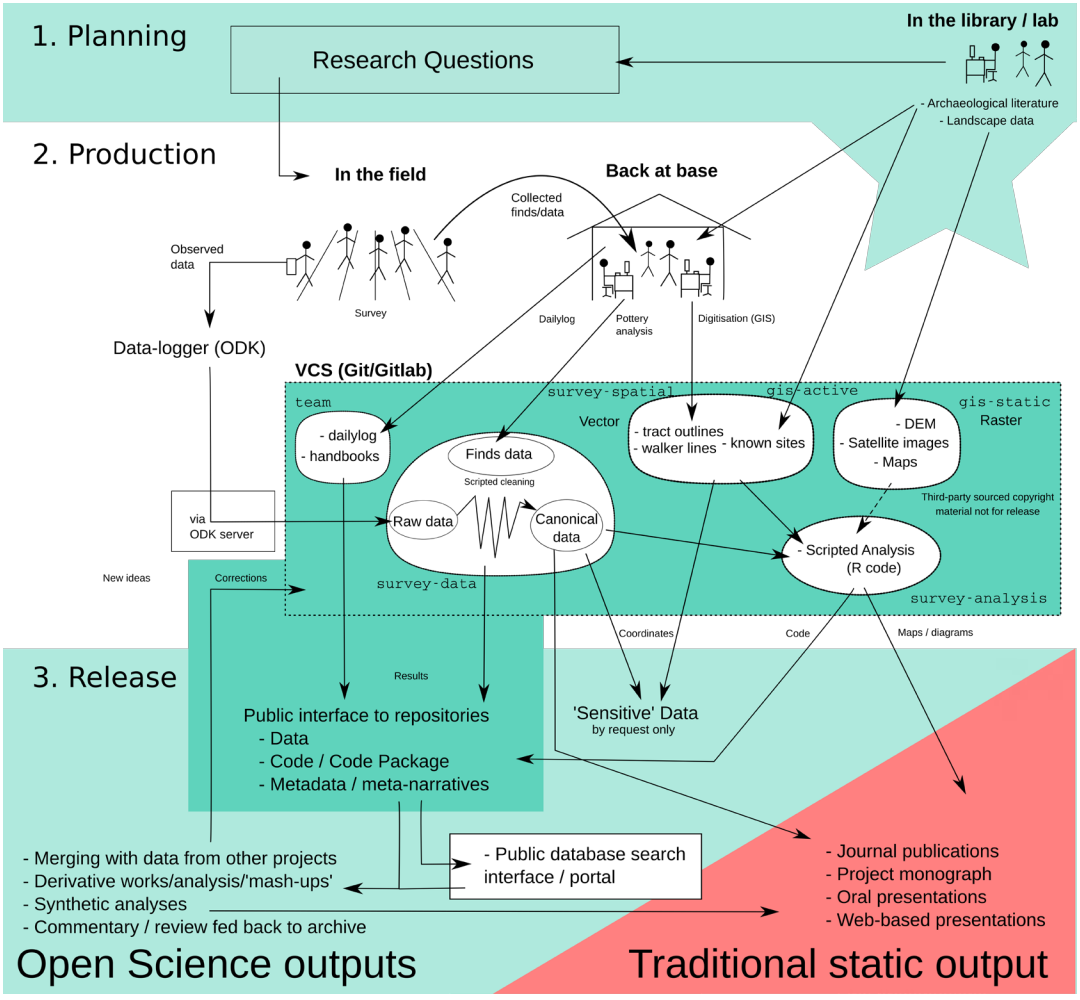

Figure 4. A simplified graphical representation of the VCS-oriented data management for Project Panormos.

Data was collected and managed in a series of stages which all relied upon the *Git*-based VCS in different ways. Whilst field-walking, the collection of data was managed by walking team leaders in two ways: 1. a paper map with a printed satellite image and grid corners to draw tract shapes, and 2. direct input of each walker's data into the data logger/GPS. *ODK* forms were filled out either at the end of each tract as field-walkers reported their finds and comments to the team-leader (`Tract`-forms) or when a particular point of interest, such as an architectural fragment or a significant viewpoint, was identified (`POI`-forms). Additionally, paper 'backups' were kept individually by each walker in form-based notebooks. The huge advantage of this system was the elimination of one tiresome and error prone procedure (namely, data-entry of paper forms) and the resultant speed by which the data could be used. The output from the *ODK* aggregate server was then stored in a *Git* repository, `survey-data`.

After an area of land had been covered, the printed maps were used as the basis to digitise tract shapes using two well-known GIS programs, *ArcGIS*'s ArcMap and *QGIS*, converted into text-oriented geographic standard format (*GML*) and then committed to the relevant *Git* repository `survey-spatial`. Meanwhile, identified finds from the survey were processed in the project house as per normal archaeological procedures. Data about each find was input into a spreadsheet and saved in CSV format. This in turn was added to the `survey-data` repository. The *RMarkdown* scripts, themselves stored in a repository and also written toward the end of the field season, drew on data about tracts, spatial location and finds to create the visualisations of find distribution (e.g. by date). The scripts could be updated almost instantly as the latest version of the data was pushed to the centralised project server, and can be easily re-used in future seasons.

The advantage of the powerful decentralised and collaborative nature of *Git* as a DVCS became very obvious. Since less time had elapsed between data collection and data visualisation than is often the case in database-oriented projects, errors in the data could easily be identified, corrected and tracked via the data history. Team members could also work on different (or indeed the same) repositories simultaneously, experimenting with various data or analytical structures, without the immediate risk of breaking others' work on the same databases. Each user could also use the most familiar software tools appropriate to the task (e.g. QGIS; ArcGIS; Vim; LibreOffice Calc; RStudio) before committing changes to the *Git* repository history using the command-line tools or GUI-equivalents (SourceTree, SmartGit).

This encouragement to compress the pipeline from data-creation via data cleaning to data analysis and the nature of *Git* architecture also had ancillary benefits for the long-term archiving of the project. Since the current tools available to manage *Git* repositories work most efficiently with plain-text-based formats, and since we were working in a cross-platform environment (different team members were using computers with Debian, Ubuntu, Windows, Mac OS X and even iOS operating systems), from the outset we were structurally encouraged to follow good data sharing and archiving practices. As has long been recognised by data archivists and advisors, plain-text formats (CSV, XML, JSON or similar) increase the likelihood of both long-term readability and re-use of data, since they are widely recognised and documented, can easily be imported by most software and, in extreme instances of loss of format protocols, are the most likely ones to remain humanly-readable and thus potentially translatable long into the future (Marwick 2016, table 1). For Panormos, the standard format for storing data from the field or finds became CSV (either exported from the *ODK Aggregate* server or spreadsheet software), and for GIS data we used *GML* (exported from the GIS software, primarily QGIS). Though each has particular limitations, both of these formats are widely recognised and recommended for archival deposit of research data (e.g. Zenodo, figshare, ADS, DANS, IANUS).

6.2 Reviewing the Approach: Open Science and Archaeological Fieldwork

Overall, we feel our pilot using the workflow outlined above was extremely successful from the point of view of encouraging better-quality collaboration using digital technology. Even those team members who were initially reluctant or dubious about the impact of combining digital data gathering using *ODK*, *Git* and *R* were pleasantly surprised by the way the application of these collaborative digital tools enhanced both practice in the field and, most strikingly, allowed us to collapse the division between field practice and the production of preliminary results with a complex data-set.

It must be admitted, however, that some of these benefits were seen because the main users of *Git* on our project (especially the current authors) had previous coding experience and were more comfortable with interacting with computers on the command line (or with coding-oriented tools) than the typical user. The application of *GitLab CE* and the encouragement of all team members to use it to produce their daily log was thus a good indicator of how typical users might interact with such complex models and points to ways that such software might become less exclusive to an emergent coding clique.

Encouragingly, all our team members were happy and willing to use *GitLab CE* to access the `team` repository and edit documents with the web pages: the basic paradigm of online editing of files (and even the use of Markdown as a basic markup format) appeared to be sufficiently familiar and intuitive for all members of whatever age (ranging from 20 to 40) or gender. There was some variation in the relative interest and willingness to learn about the underlying structure of the software and the motivation behind using it (in an admittedly small sample, most interest came from younger male team members, which also reflected a general interest in computing in archaeology). To us, this suggests that more education about reproducibility, computing and data science is needed in archaeological syllabi. A more data-science or archaeological user-interface to *Git*, which foregrounds the functions that the majority of archaeologists really need, would also help: DVCS can create highly complex structures, but that does not mean that most end-users need to see that complexity. In the same way that Wikipedia or Facebook presents an easy-to-use interface to a highly complex relationship model, new online tools (perhaps adapted from open-source interfaces like *GitLab CE*) could enhance the experience such that its use could be more easily inserted into everyday archaeological field workflows.

7 Some Practical Limitations of *Git*: Large-files and Internet Reliability

The systemic qualities and particular strengths of *Git* naturally guided us toward certain usage patterns, some of which are beneficial in an archaeological environment and some of which, we might argue, are potentially detrimental. Given *Git*'s origin as a tool to version control source code, it is optimised and most useful for plain-text and struggles to manage the version history of non-textual (i.e. binary) data. Thus, the standard edition of *Git* encourages the storage of small plain-text data formats, excellent for long-term preservation, but restricting the types of data that can be efficiently versioned. This was clearly the case when it came to managing Project Panormos' most important binary assets, namely photographs, as well as large GIS raster images (such as satellite imagery) which were needed for visualisation.

In response to this limitation, non-standard but more complex extensions have been developed to manage large binaries. The two most popular include *Git Large File Storage* (developed by GitHub), and *git-annex* (developed by Joey Hess), support for which was integrated into versions of *GitLab CE* after the pilot was begun, but which requires the installation of additional software on both the server (to enable access and control of the files via the *GitLab CE* server) and client machines (on each user making direct use of *Git*). These extensions are particularly apposite in cases where large binaries need to be kept together in sync with other text-oriented versionable data (e.g. source code and fixed but relatively unchanging binary assets). However, the tools to manage changes between binary versions are still limited (each version is stored in its entirety rather than a more efficient record of changes, called deltas, which is how *Git* acts on text or code). Although we initially continued to use a *Git* repository to back up and sync the large raster assets for the short-term as a matter of platform convenience, it became clear that *Git* was not a good platform for maintaining the photographs, especially because the images themselves did not require version control, since we did not expect to edit the files. The only useful function of versioning here would be an archive history of when images were entered into the repository so that users could identify if they had the full database or not.

A further problem with using *Git* with large files was the limitations imposed by the internet connection in the field. We found that in practice even separating the images into a separate `photo-archive` repository was impractical because our internet connection was too weak to enable uploading of large commits. The

connection would simply time-out before *Git* had been able to upload a large number of changes to the server. This kind of problem might be offset, to some degree, by the use of local 'staging' servers within a local field-based network, which could periodically synchronise with the off-site centralised server (e.g. during the night), or the use of *Git-LFS,* which manages large files more sensibly. In essence, alternative methods of integrating version control with large binary assets such as archaeological images would need a more complicated IT infrastructure than we had set up in our relatively small-scale survey operation. Most important for both photograph and GIS assets is syncing of the most up-to-date archive (without necessarily versioning changes), options for which we are continuing to explore.

7.1 Attributing Work to Authors

The *Git* model is based on the assumption that collaborators are independent and differentiated, but interact in relatively similar ways with the code (i.e. on the computer). Diagrams that allow reviewers to trace responsibility for changes to certain contributors are thus easy to construct in *Git*. This is useful to trace bugs, for example, but also to recognise the degree to which different users contribute to the project. We had hoped that the use of *Git* version control would enable a greater deal of recognition of work done before publication by tracking contributions to the data sets, but *Git* is not designed to document the effort expended in the sort of non-computer based tasks which are essential to archaeological work (e.g. find drawing, washing ceramic, cooking). Additionally, in the absence of obvious easy-to-use tools and in light of the issues above regarding bandwidth, it cannot easily track some almost exclusively digital workflows (e.g. digital photos). Because many of our team members were uncomfortable using *Git* at the command line (or even indirectly through GUIs), only a few of us performed the `commit` and `push` actions on data. However, the resulting tag in the *Git* version history was still an accurate reflection of who had actually done the work because *Git* differentiates between 'authors' and 'committers'. For both prose and tabular data, the way *Git* tracks changes, i.e. by line, can cause issues with how smaller changes can be visualised. For example, small edits (e.g. changing the content or one cell in a table or adding or deleting a full stop or word) are tracked as full line changes, making review harder than the equivalent tracking tools built into many word-processing tools.

What all this means in practice is that the commit history cannot be used as an accurate metric for project contribution because it would skew toward those involved in data-entry and management over those actually producing the data in non-digital ways. Instead, the *Git* history provides a powerful way of auditing of mistakes.

7.2 Data Cohesion, Data Models and Databases in DVCS

As a version-control system designed primarily to manage code development, *Git* tries to be as content-agnostic as possible. As such, it necessarily makes no attempt to monitor the semantic cohesion of the repository's content. Nonetheless, in a coding context, its ability to keep a level of cohesiveness by tagging of versions (and hence syncing different parts of a code project) is often emphasised as one of its strengths for collaborative work.

Scientific data often has similar requirements for cohesion, especially where different data sets need to be linked together for comparison or analysis. In the digital age, the most proficient archaeological projects have put such data into relational databases (*MySQL*, *Access*, and other SQL-type database) using 'keys' (or 'IDs') to link data between different tables through querying. Unfortunately, *Git* is not an ideal platform for versioning such relational databases and it certainly does not aim to replace them. Instead, tabular data must be kept in text-formats such as CSV, or XML, and this has issues for scalability, cohesive editability and problems standardisation of data for which relational databases were precisely designed to overcome. A partial solution to this limitation is to continue to use relational databases for live editing while depositing the data periodically into a *Git* repository via export to CSV or similar. Alternatively (or perhaps even in parallel), another solution is to use the automated 'continuous integration' features of platforms like *GitLab CE* (which was made available after our pilot and which we are currently testing), designed originally to

continually test code, to instead check the referential integrity of data stored in CSV files by import to a relational database whenever changes are made to the data. This is effectively a stop-gap for alternative version-control-systems that can manage tabular data better than *Git*, but the important point is that the use of *Git* in data management workflows should not somehow be seen as a replacement for good database design.

7.3 Sustainability of Data

For long-term sustainability, it is good to ask what would happen with the Panormos data in 100 years when the world will probably have forgotten about *ODK*, *Git*, *GitLab CE* and *R*. Of course, if the repositories are not migrated periodically into up-to-date formats, then there is a potential for information loss: tracing the authorship of commits may be difficult (*Git*), the web interface will no longer work (*GitLab CE*) and the code for the analysis (*R*) may not be possible to run on current hardware. However, one of the main advantages of a working *Git* repository is that the history is stored in a different, hidden folder (`/.git`) and the latest version of the data is preserved in its latest state in a standard file-system. This means that as long as good archival formats are selected and the file system used is robust, the data has the best chance of survival. Additionally, in the shorter-term, *GitLab CE* relies on very widely supported Free Software platform (Unix, Ruby and Git) and has its own API, which means that the content of the *Git* repositories can be easily transferred and duplicated.

From the point of view of the *R* code, the benefit of the prose-oriented 'literate programming' that *RMarkdown* documents encourage is the production of a script designed to be understandable to computers AND humans, such that even if the reproducible analysis is broken or the code interpreter obsolete, the commentary which accompanies the code should be sufficiently self-explanatory to be translated into a new language.

The design of the Project Panormos Survey data management therefore strengthens its long-term preservation and scope for reuse. The minutiae of the author-tracking system may be lost forever at some point, but this will no doubt be less important when the data have already been checked, analysed and re-used. The data itself belongs to the public domain and the careers of the archaeologists involved are therefore no longer relevant.

8 Ethical Issues Raised by Open Science Transparency

Open Science, transparency, and reproducible research are not, in a sense, totally novel, even if such keywords may be needed to drive the needed transformation in the dissemination of knowledge in the archaeological world which can help us to deal with the growing 'data deluge' (Bevan 2015). The transition to an increasingly digital workflow in fieldwork practice, as in publishing paradigms, presents both many new risks and new opportunities, some of which we want to explore further here with the experience and knowledge gained from our pilot. In the context of the increasing currency of openness and transparency as both principles and brands within academia, as well as in the wider public sphere (cf. Götz & Marklund 2014), much less attention has been paid to certain unintended effects of transparency, especially with regard to the protection of vulnerable archaeological sites and the protection and nurturing of archaeological talent and careers. Part of the solution to these problems may come from addressing particular modes of transparency which are best suited to the kind of work that archaeologists do.

8.1 Transparency and Looting: Protecting Sites and Heritage?

One under-discussed issue that we were particularly concerned about with regards to Open Science approaches was the protection of cultural heritage from looting and vandalism. The principle of free data and analysis naturally encourages archaeologists to release the geographical location of their finds as a key part of their understanding. This would seem to be especially important for a survey where spatial location

is the only contextual data available. The problem, of course, is that freely releasing GIS coordinates online could potentially help looters and illegal excavators find sites as well as *bona fide* archaeologists. Of course, maps published in paper form have long presented the same danger, but until recently these documents were difficult to access and GPS technology was highly restricted. In the context of digital circulation of academic articles and the incorporation of GPS devices within most mobile telephones, it is not clear the extent to which the publication of geographical coordinates online has or will inflate the increasing scale – arguably to an industrial level – of non-archaeological excavation and looting across the world (Costa, Beck, Bevan, & Ogden 2013, 453). In Turkey, illegal excavation is an acknowledged problem,[3] and evidence for it can be seen at many locations.[4] Amongst professional and academic archaeologists, we have encountered essentially only two attitudes: one which says that publishing coordinates is irresponsible and another which argues that the only people hindered by holding back coordinates are archaeologists (because local residents, who are often seen as the primary agents of the looting, have far better knowledge of the location of the archaeological finds). To our mind, the main problem is the lack of evidence for either view, which means the in-built assumptions of both are very difficult to test. Unsurprisingly, many project groups, including ourselves, are disinclined to be a test case! It is fair, we believe, to publish the general coordinates of large and protected sites when they are already known to locals or the general public. Indeed, attempting to stop this would be futile, given the number of geographic tags relating to ancient sites to be found in the publicly accessible Google Earth or Open Street Map databases. We must simply encourage and support state institutions to protect the sites sufficiently.[5] But for small-scale survey scatters or tiny, remote or difficult to protect sites, the ethics of open online access to their location is complex, and state institutions view their publication with considerable reserve. Once published, of course, it is almost impossible to withdraw and control digital data.

8.2 Transparency and Scooping: Protecting Archaeologists?

One downside of speedy open publication of raw data direct from the field is its potential amplification of already-existing power relationships within the academy. In a career environment in which there is little job security and a decreasing number of permanent posts compared to the number of junior researchers, the danger is that those who already have access to funds and infrastructure can make use of the data that more junior and less established researchers have produced before the producers have had a chance to harvest the normal set of publications (i.e. the data is 'scooped'). Hence, greater academic credit accrues to data crawlers or data scrapers than to the producers. At the time of planning, all of those involved with the Panormos Survey were junior or consolidator level without permanent posts, and so this was something we considered seriously in establishing the nature of our Open Data policies. Whether or not such fears are justified is, again, difficult to assess. When there may be legal issues involved in situations of alleged or confirmed idea/data theft and scooping, much like the ethics of coordinate publication, discussion involving empirical data on this issue has been rare, instead confined predominantly to informal speculation of 'what ifs' or anecdotal experiences.

Based on an experiment in keeping an open online lab notebook, Carl Boettiger, who works in the field of computational ecology, suggested that the benefits were greater than this so far unproven risk (Gewin 2013). This kind of open lab notebook approach works particularly well for experimental and theoretical work that does not involve living subjects. It was something we piloted in the form of the daily log to capture the atmosphere of the project. We were, however, uncomfortable with making

3 See, for example, the reports produced by the TAY project: http://www.tayproject.org/.

4 Some cases are listed at http://arkeolojihaber.net/tag/kacak-kazi/ . See also the list of the most important pieces looted the last decade, found in foreign countries and brought back to Turkey http://www.kulturvarliklari.gov.tr/TR,44751/2004--2015-yillari-arasinda-ulkemize-iadesi-saglanan-es-.html (May 18, 2017).

5 When the state breaks down, however, as it has recently in Afghanistan, Iraq and Syria, the consequences of industrial looting are well known (American Association for the Advancement of Science 2016).

entries publicly accessible on the web pages without vetting and editing. Besides wishing to protect archaeological sites from harm by avoiding publicity, we were additionally uncertain about the ethical implications of referring to team members and the attitude of local authorities to such transparency. Archaeological fieldwork, especially where it is conducted in a country other than one's own, often rests on careful diplomacy, personal relationships and tact. These dynamics can be at odds with the open publication of material that Open Science appears to demand.

For data, a popular approach is to make cleaned source data available at the moment or unit of the publication, e.g. to attach the data in the form of an annex to a particular journal paper or on another form of public server (e.g. Zenodo, GitHub, ADS, an institutional repository or a private web page, etc., see Table 3) when or soon after the final 'results' are published. The advantage is that it provides confidence that the data is "clean, valid and meaningful" (Marwick, pers comm.) and to protect authors from "the worst case of scooping, where an unaffiliated scholar is the first to publish substantial findings from our data." Though providing protection to the primary scholar, we feel this fails to address one of the major problems in archaeological fieldwork, namely the frequency with which archaeological progress is held back by the unwillingness or difficulty with which many finds are published in citable form before the final publication (which because of the difficulty of pulling together such a monumental document, too often never makes it to print). Such practices of data-hoarding are extremely common in certain parts of the archaeological world; in the worst cases, the result is obviously that much knowledge is lost in a long process, which makes it difficult to correct mistakes and feed critique back into field methodologies before it is too late. How can as much archaeological documentation be made public as possible whilst maintaining the credit for creating the data in the first place? Besides this, making only 'cleaned' data available at the 'end,' however defined, also obscures the process of 'cleaning.' One of the aims of Open Science is to examine, critique and contribute back into the whole workflow.

The ideal solution, of course, is an academic environment in which primary research contribution and open data publication are more appropriately recognised. There are nascent forays into raising the prestige of data publications: the *Journal of Open Archaeological Data* provides a peer-reviewed outlet and implementation of this idea in which the analysis of the datasets is kept to a minimum. But undeniably the limited number of contributed papers so far (22 data papers since foundation in 2012, as of April 2017) compared to the countless archaeological projects being undertaken across the world reflects the fact that prestige still adheres to older formats, or at least established journals, in a market in which there are a confusing plethora of specialist publications. The model of such journals is also inherently static: credit accrues to the compiler of the database as a static entity, the 'paper', but what happens when new finds demand an update? An addition of one find may be important, but would it justify an entirely new 'data paper'?

In the absence of a fundamental shift in systems of academic progression, licensing (i.e. legal protection of intellectual property) remains the main avenue by which the risk of scooping can be managed. This is how the Free Software movement functions (The Free Software Foundation 2016), and there are equivalent licenses such as Creative Commons, Open Data Commons (see table 4) which are appropriate for scientific data (Open Definition 2016). This involves attaching meta-data on the reuse of published data and analysis for a certain duration: for example, by specifying a license allowing access and browsing but no 'derivative work' or re-publication during a defined embargo. In concrete terms, this would involve publishing first under a Creative Commons CC BY-ND license, followed by re-release under a CC BY license after a specified time period. If a five-year barrier on reusage is placed upon initial primary data publication, this would provide a reasonable time for the primary researchers to establish a secondary publication trail, with the additional bonus that outside scrutiny might provide contributions that improve the secondary analysis. Once all the contributors are satisfied, the restriction might also be lifted earlier than five years.

9 Open Science Modalities: Current Project Panormos Survey Policies

A sensible ethical distinction can thus be made between two or more different *modes* of Open Science practice. One involves production and release as more-or-less simultaneous, i.e. *instant release*. This is an idealist's version, but one which is often problematic, especially where certain types of data must be protected ethically and could lead to harm. In another *delayed release*, public access is delayed until potential ethical, technical or 'scooping' issues are smoothed or checked. For looters, licensing does not protect sites from harm. In both cases, what is essential is that the reproducibility of workflows, data manipulation and analysis are recorded and therefore made transparent by the point of release or, where ethical issues still remain, that systems are in place to allow *bona fide* reviewers access to the full data history. Such systems are unlikely to be as complex as those used for medical or social data about human subjects (unless they involve indigenous rights issues), but the digital infrastructure needs to be carefully thought through nonetheless. In this second mode, it is also important to establish conventions for the community regarding the reasonable maximum on the difference between production and release to prevent a re-inscription of old and damaging habits of primary data hoarding.

For the Project Panormos Survey, we decided to fix a maximum delay of 24 months between survey data production and an initial restricted data release, with a five-year embargo on derivative scientific works from the end of that season's work. For example, this means that the data from the 2015 pilot would be definitively available online to read in working form by October 2017 at the latest, and then available for reuse by non-project members by October 2020 at the latest. Besides the tract and find data (derived from the `survey-data` repository), we also plan to release project handbooks and daily logs (derived from the `team` repository) with a less restrictive re-use license. We hope this will encourage (with due attribution) derivative modulations of our methodologies both in the field and in terms of the collaborative platform (i.e. software).

In deference to the unresolved questions around the necessity to protect the spatial location of archaeological remains, we have opted to keep back real geographic data (i.e. coordinates showing tract locations and points of interest, which are stored in `survey-spatial`), and instead publish blank surrogates to facilitate oversight of methods and computational reproducibility. Access to the real data will be granted by request to *bona fide* researchers and heritage protectors to replace these surrogates.

Our original plan had been to make all of our *Git* repositories available for public access, but we did not fully realise the consequences of the fact that *ODK* embeds GPS data within the exported field form data files. As such, it would be impossible to open our original raw *Git* archive without breaking the imperative to protect spatial data. For our purposes, we are therefore forced to create a 'public' version of `survey-data` with the spatial data redacted. This is clearly unsatisfactory within the spirit of full transparency. We would advise future projects who use version control to plan for these kinds of eventualities very carefully when committing data to data histories. Projects should also pay attention to discussion around the similar security issues that have been noted in the code development world, e.g. the prevalence of passwords stored (and therefore accessible) in the public *Git* archives of Open Source projects (Lagoze et al. 2013).

10 Concluding Comments: Breaking the Hegemony of the 'Final Report'

As we have demonstrated with our pilot project, new software is beginning to offer field practitioners a practical set of tools that can considerably increase the transparency of the data-collection process and also bring other benefits of an Open Science approach to archaeology. The particular platforms we piloted could be replaced by others, and there are functionality and educational gaps specific to archaeology which make the universal uptake of our approach slow in the very near future. Our feeling, however, was that it is simply a matter a time until many more projects will make version-controlled data management central

to their approach. We have tried to acknowledge and manage some of the objections to an Open Science approach, as well as highlight the limitations and caveats which must be considered by others attempting a similar pilot to ours.

Overall, we feel that the benefits of our approach are more numerous than the risks. The process of exposure of the field-to-shelf pipeline, warts-and-all, enables archaeologists not only to show working and hence enhance the ability to assess final conclusions, but also to increase the reusability of data and enable *alternative* conclusions. In this, we feel that version-history will become the most important tool in the rejecting the hegemony of the traditional research product of archaeology, namely the 'final fieldwork report', which falsely presents itself as unproblematically finished or complete. Enabling and encouraging archaeologists to publish (or make public) data which is unfinished or unfinalised offers the opportunity for greater transparency, greater collaboration, fewer lost contexts and hidden, especially negative, results. We hope that our pilot will offer inspiration for further research and analysis using an Open Science approach; and perhaps most pressingly from an archaeological perspective, help discourage the sort of sad data loss caused by the languishing of unpublished finds and contexts.

Project repository: For the medium-term public access to data, a redirecting URL has been set up, which will forward interested readers to the current Project Panormos Survey data repositories and/or releases as they appear: http://reproducible.panormos.de/data/. General information and updates about the project including an up-to-date publication list will continue to be available from the main project website: http://www.projectpanormos.com/.

Acknowledgements: Thanks to: Anja Slawisch (DAI), Hasibe Akat (Milet Müzesi), as directors of the Project Panormos field projects from 2012 to 2015; Sebastian Weber (DAI) for technical help; the whole 2015 Panormos Survey team for their enthusiastic and witty contributions to the experiment, including daily log entries. The survey project was funded in 2015 by the German Archaeological Institute (DAI). In the spirit of archaeological collaboration, Anja Slawisch (DAI) and Ben Marwick (Washington) offered invaluable advice and suggestions to early drafts of this article, which considerably improved to the manuscript. Additional valuable suggests were made by two anonymous referees to whom we are very grateful. Remaining errors naturally remain at the authors' doors.

References

Alcock, S., & Cherry, J. (Eds.). (2004). *Side-by-side Survey: Comparative Regional Studies in the Mediterranean World*. Oxford: Oxbow.

American Association for the Advancement of Science. (2016). Ancient History, Modern Destruction: Assessing the Status of Syria's Tentative World Heritage Sites Using High-Resolution Satellite Imagery. https://www.aaas.org/page/ancient-history-modern-destruction-assessing-status-syria-s-tentative-world-heritage-sites-7 (May 18, 2017).

Anokwa, Y., Hartung, C., Brunette, W., Lerer, A., & Borriello, G. (2009). Open Source Data Collection in the Developing World. *IEEE Computer*, 97–99. http://doi.ieeecomputersociety.org/10.1109/MC.2009.328.

Bartling, S., & Friesike, S. (Eds.). (2014). *Opening Science. The Evolving Guide on How the Internet is Changing Research, Collaboration and Scholarly Publishing*. Springer. https://doi.org/10.1007/978-3-319-00026-8.

Bazerman, C. (1983). Scientific Writing as a Social Act: A Review of the Literature of the Sociology of Science. In Paul V. Anderson R. John Brockmann & C. R. Miller (Eds.), *New essays in technical and scientific communication: Research, theory, practice* (pp. 156–184). Amityville: Baywood.

Bevan, A. (2012). Value, Authority and the Open Society. Some Implications for Digital and Online Archaeology. In C. Bonacchi (Ed.), *Archaeology and digital communication: Towards strategies of public engagement* (pp. 1–14). London: Archetype.

Bevan, A. (2015). The data deluge. *Antiquity*, *89*, 1473–1484. https://doi.org/10.15184/aqy.2015.102.

Bissell, M. (2013). Reproducibility: the Risks of the Replication Drive. *Nature*, *593*, 333–334. https://doi.org/10.1038/503333a.

Bowers, J. (2011). Six steps to a Better Relationship with Your Future Self. *The Political Methodologist*, *18*(2), 2–8.

Chacon, S., & Straub, B. (2014). Pro Git. https://git-scm.com/book/en/v2 (May 18, 2017).

Costa, S., Beck, A., Bevan, A., & Ogden, J. (2013). Defining and advocating open data in archaeology. In G. Earl, T. Sly, A. Chrysanthi, P. Murrieta-Flores, C. Papadopoulos, I. Romanowska, & D. Wheatley (Eds.), *Archaeology in the Digital Era*.

Papers from the 40th Annual Conference of Computer Applications and Quantitative Methods in Archaeology (CAA), Southampton, 26-29 March 2012 (pp. 449–456). http://dare.uva.nl/document/516092.

Creative Commons (2013). Responses of Creative Commons to House of Lords Comments Concerning the CC BY License. https://wiki.creativecommons.org/wiki/BIS_committee_UK_OA_Policy#Plagiarism (May 18, 2017).

Düring, B., & Glatz, C. (Eds.). (2016). *Kinetic Landscapes. The Cide Archaeological Project: Surveying the Turkish Western Black Sea Region*. Warsaw/Berlin: De Gruyter Open. https://doi.org/10.1515/9783110444971.

Ellis, S. J. (2016). Are We Ready for New (Digital) Ways to Record Archaeological Fieldwork? A Case Study from Pompeii. In E. W. Averett, J. M. Gordon, & D. B. Counts (Eds.), *Mobilizing the past for a digital future: The potential of digital archaeology* (pp. 51–75). Grand Forks: The Digital Press at the University of North Dakota.

Ersoy, Y., & Koparal, E. (2008). Urla ve Seferihisar İlçeleri Yüzey Araştırması 2007 Yılı çalışmaları. *Araştırma Sonuçları Toplantısı*, *26*(3), 73–9.

Ersoy, Y., Tuna, N., & Koparal, E. (2010). Urla ve Seferihisar İlçeleri Yüzey Araştırması 2009 Yılı Çalışmaları. *Araştırma Sonuçları Toplantısı*, *28*(2), 339–360.

EU. (2016). H2020 Programme – Annotated Model Grant Agreement (Version 2.1.1). https://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/amga/h2020-amga_en.pdf (May 18, 2017).

Fanelli, D. (2013). Redefine misconduct as distorted reporting. *Nature*, *494*, 149. https://doi.org/10.1038/494149a.

Fitzpatrick, K. (2011). *Planned obsolescence: Publishing, technology, and the future of the academy*. New York: New York University.

Germán, D. M., Adams, B., & Hassan, A. E. (2016). Continuously mining distributed version control systems: An empirical study of how linux uses git. *Empirical Software Engineering*, *21*(1), 260–299. https://doi.org/10.1007/s10664-014-9356-2.

Gewin, V. (2013). Turning Point: Carl Boettiger. *Nature*, *493*, 711. https://doi.org/10.1038/nj7434-711a.

Götz, N., & Marklund, C. (2014). *The Paradox of Openness: Transparency and Participation in Nordic Cultures of Consensus*. Leiden: Brill.

Harley, D., Acord, S. K., Earl-Novell, S., Lawrence, S., & King, C. J. (2010). Archaeology Case Study. In *Assessing the Future Landscape of Scholarly Communication: An Exploration of Faculty Values and Needs in Seven Disciplines*. (pp. 29–136). UC Berkeley: Center for Studies in Higher Education. http://escholarship.org/uc/item/15x7385g.

HEFCE. (2015). *Policy for open access in the post-2014 Research Excellence Framework: Updated July 2015*. Higher Education Funding Council for England. http://www.hefce.ac.uk/pubs/year/2014/201407/(May 18, 2017).

Heller, L., The, R., & Bartling, S. (2014). Dynamic Publication Formats and Collaborative Authoring. In S. Bartling & S. Friesike (Eds.), *Opening science* (pp. 191–211). Springer International Publishing. https://doi.org/10.1007/978-3-319-00026-8_13.

Hodder, I. (1999). *The Archaeological Process: An Introduction*. Oxford: Blackwell.

Hodder, I. (2008). Evaluating Multiple Narratives: Beyond Nationalist, Colonialist, Imperialist Archaeologies. In J. Habu, C. Fawcett, & J. M. Matsunaga (Eds.), *Evaluating multiple narratives: Beyond nationalist, colonialist, imperialist archaeologies* (pp. 196–200). New York: Springer New York. https://doi.org/10.1007/978-0-387-71825-5_13.

Huggett, J. (2015). Digital Haystacks: Open Data and the Transformation of Archaeological Knowledge. In A. T. Wilson and B. Edwards (Ed.), *Open source archaeology* (pp. 6–29). Warsaw/Berlin: De Gruyter Open. https://doi.org/10.1515/9783110440171-003.

Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299–314.

Journal of Open Archaeology Data. (2017). Research integrity. http://openarchaeologydata.metajnl.com/about/research-integrity/ (May 25, 2017).

Kansa, E. (2012). Openness and archaeology's information ecosystem. *World Archaeology*, *44*(4), 498–520. https://doi.org/10.1080/00438243.2012.737575.

Kansa, E. (2014). The Need to Humanize Open Science. In S. A. Moore (Ed.), *Issues in open research data* (pp. 31–58). London: Ubiquity Press. https://doi.org/10.5334/ban.c.

Kansa, E., & Whitcher Kansa, S. (2013). We All Know That a 14 Is a Sheep: Data Publication and Professionalism in Archaeological Communication. *Journal of Eastern Mediterranean Archaeology and Heritage Studies*, *1*(1), 88–97. https://doi.org/10.1353/ema.2013.0007.

Kansa, E., Whitcher Kansa, S., & Arbuckle, B. (2014). Publishing and Pushing: Mixing Models for Communicating Research Data in Archaeology. *International Journal of Digital Curation*, *9.1*, 57–70. https://doi.org/10.2218/ijdc.v9i1.301.

Kelty, C. M. (2005). Free Science. In J. Feller, B. Fitzgerald, S. A. Hissam, & K. R. Lakhani (Eds.), *Perspectives on free and open source software* (pp. 415–430). MIT Press.

Kelty, C. M. (2008). *Two Bits: The Cultural Significance of Free Software*. Durham: Duke University Press. http://twobits.net/pub/Kelty-TwoBits.pdf.

Kelty, C. M. (2014). Beyond Copyright and Technology: What Open Access can tell us about Precarity, Authority, Innovation, and Automation in the University Today. *Cultural Anthropology*, *29*(2), 203–215. https://doi.org/10.14506/ca29.2.02.

Knorr, K. D., & Knorr, D. W. (1978). *From Scenes to Scripts: On the Relationship between Laboratory Research and Published Paper in Science*. Vienna: Institute for Advanced Studies.

Knuth, D. E. (1984). Literate Programming. *The Computer Journal*, *27*(2), 97–111. https://doi.org/10.1093/comjnl/27.2.97.
Lagoze, C., Block, W. C., Williams, J., Abowd, J., & Vilhuber, L. (2013). Data management of confidential data. *International Journal of Digital Curation*, *8*(1), 265–278.
Lake, M. (2012). Open archaeology. *World Archaeology*, *44*(4), 471–478. https://doi.org/10.1080/00438243.2012.748521.
Latour, B., & Woolgar, S. (1979). *Laboratory Life: The Social Construction of Scientific Facts*. Beverley Hills: Sage.
Marwick, B. (2016). Computational Reproducibility in Archaeological Research: Basic Principles and a Case Study of Their Implementation. *Journal of Archaeological Method and Theory*, 1–27. https://doi.org/10.1007/s10816-015-9272-9.
Masic, I. (2012). Plagiarism in scientific publishing. *Acta Informatica Medica*, *20*(4), 208–213. https://doi.org/10.5455/aim.2012.20.208-213.
Matthews, R., & Glatz, C. (Eds.). (2009). *At Empires' Edge. Project Paphlagonia Regional Survey in North-Central Turkey*. London: British Institute at Ankara.
Morin, A., Urban, J., Adams, P. D., Foster, I., Sali, A., Baker, D., & Sliz, P. (2012). Shining Light into Black Boxes. *Science*, *336(6078)*, 159–160. https://doi.org/10.1126/science.1218263.
Nielsen, M. (2011). An informal definition of openscience. http://www.openscience.org/blog/?p=454 (May 18, 2017).
Nuzzo, R. (2015). How scientists fool themselves – and how they can stop. *Nature*, *526*, 182–185. https://doi.org/10.1038/526182a.
Open Definition. (2016). Conformant license (version 2.1). http://opendefinition.org/licenses/ (May 18, 2017).
R Core Team. (2017). R: A Language and Environment for Statistical Computing. https://www.R-project.org (May 18, 2017).
Schloen, D. (2001). Archaeological Data Models and Web Publication Using XML. *Computers and the Humanities*, *35*(2), 123–152. http://www.jstor.org/stable/30204847.
Schloen, D., & Schloen, S. (2014). Beyond Gutenberg: Transcending the Document Paradigm in Digital Humanities. *Digital Humanities Quarterly*, *8*(4). http://www.digitalhumanities.org/dhq/vol/8/4/000196/000196.html.
Smith, J. (2013). Adapting Git for simple data. https://theodi.org/blog/adapting-git-simple-data (May 14, 2017).
Strupler, N. (2016). Archaeology as Community Enterprise. In S. Campana, R. Scopigno, G. Carpentiero, & M. Cirillo (Eds.), *CAA2015 keep the revolution going, proceedings of the 43 rd annual conference on computer applications and quantitative methods in archaeology* (Vol. 1, pp. 1015–1018).
The Free Software Foundation. (2016). Various Licenses and Comments about Them. https://www.gnu.org/licenses/license-list.en.html (May 18, 2017).
Vinck, D., & Clivaz, C. (2014). The Humanities Unbound. Knowledge and Culture Reinvented Outside the Book. *Revue d'anthropologie des connaissances*, *8*(4), a–w. https://doi.org/10.3917/rac.025.0682.
Wallis, J., Rolando, E., & Borgman, C. (2013). If We Share Data, Will Anyone Use Them? Data Sharing and Reuse in the Long Tail of Science and Technology. *PLoS ONE*, *8*(7), e67332. https://doi.org/10.1371/journal.pone.0067332.
Wallrodt, J. (2016). Why Paperless: Technology and Changes in Archaeological Practice, 1996–2016. In E. W. Averett, J. M. Gordon, & D. B. Counts (Eds.), *Mobilizing the past for a digital future: The potential of digital archaeology* (pp. 33–50). Grand Forks: The Digital Press at the University of North Dakota.
Xie, Y. (2015). *Dynamic Documents with R and knitr* (2nd ed.). Boca Raton, Florida: Chapman; Hall/CRC.
Xie, Y. (2016). *Knitr: A General-Purpose Package for Dynamic Report Generation in R* (R package version 1.12). http://yihui.name/knitr/.

Appendix

OpenDataKit (ODK)

OpenDataKit is a suite of software and related protocols designed to facilitate a complete flow of form design, data entry and data aggregation primarily organised around the semi-Open Source Android platform. Form templates are stored in *XForm* syntax. The forms can include multiple choice questions, free text input, dates and media such as photographs. We used *XLSForm* to design our forms, which relies on a simple Excel-table structure to manage questions and expected answers, and kept versioned copies of this as it evolved in the Project's *GitLab CE*-hosted `survey-design` repository. The latest version of the form was converted into an *XForm* and uploaded to our central *ODK Aggregate* server for the project. In our case we used a Google-based service called *App Engine*, to run *ODK Aggregate* but the software can be installed on many other types of server. Once set-up individual mobile devices can be set up to download the forms through an app called *ODK Collect*. This is essentially a data-entry interface that allows the user to fill in multiple instances of this form and then can manually or automatically upload the results to the server.

Git

Git was originally designed to handle the development of the Linux kernel but which has now become the most widespread VCS system for collaborative coding across the internet. The system works by assigning the status of 'git repository' to a particular directory or folder within the computer. After this, any files in the folder can be committed (both saved and tagged) into a special hidden directory that *Git* maintains as the repository's history: effectively preserving a 'snapshot' of all the files in the repository with details of who made the snapshot and when. Once a set of tasks have been achieved, the repository (and all its changes) can then be 'pushed' (uploaded) to another repository across the internet (either someone else's computer, or, more frequently, a central project server). From there, others can review changes and also push their own changes, which can be complementary or contradictory to the other person's commits, but in order to continue conflicting changes must be resolved by selecting particular versions to go forward. Git offers some very powerful tools for tracking different branches of a project to avoid data loss, though it is not an automatic system of synchronization and requires active intervention by users to maintain 'clean' data histories.

A Vignette for Using Git in Archaeology Research (Figure 5)

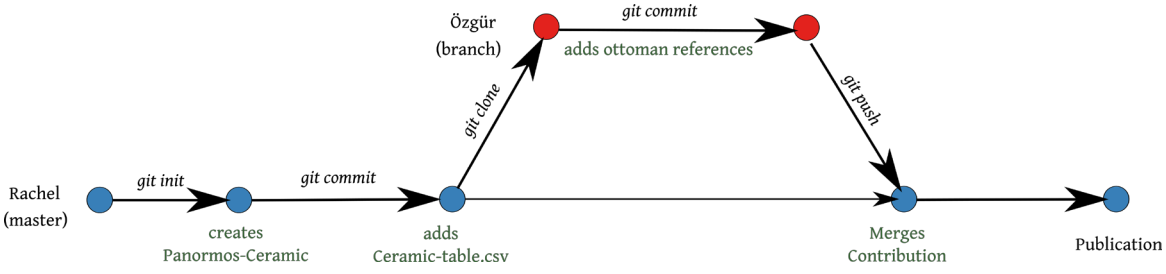

Figure 5. A simplified graphical representation of a Git-workflow.

Rachel[6] decides to start a project about an assemblage of ceramics found near Panormos. She makes a new folder on her computer and calls it 'Panormos-Ceramic'. With *Git* (either using the command line or a graphical interface) she transforms her folder into a *git repository* (`git init`). Now, every snapshot made

6 Rachel and Özgür are fictional characters, but willing to play in this fiction. As in all fairy tales, this story simplifies the process a little for clarity. A version of this story was used to explain *Git* on the Project's team website.

inside this folder is tracked (a snapshot, in 'git-language', is called a *commit*, saves the state of the project)[7]. She creates a first table, called 'Ceramic-table.csv', using her normal table software, with all the information she could find in the literature and saves her changes with a snapshot (`git commit`). During her research at the library, she meets Özgür, a specialist on the Ottoman period. Özgür decides to collaborate with Rachel and *clones* (i.e. makes a copy of) the 'Panormos-Ceramic' repository (`git clone`). Özgür edits his copy of the table 'Ceramic-table.csv' with references from Ottoman literature that Rachel couldn't read. As soon as he finishes his work, Özgür *commits* his copy of the table (i.e. saves a snapshot in his 'Didyma-Ceramic' repository) (`git commit`). Now Özgür's repository contains his changes and the initial data from Rachel, knitted together. *Git* tracked the time of the changes and assigned the name of the author to each line of the table. Rachel and Özgür know that their work is attached with their name. Özgür sends his changes to Rachel by uploading or *pushing* his copy back to Rachel's computer or a shared central server (`git push`). When Rachel sees the changes, she is really happy and *merges* Özgür's changes into her repository (`git merge`). A month later, Rachel finishes her work and wants to publish her results. With the help of *Git* it is now easy to acknowledge Özgür's contribution.

GitLab CE

GitLab as a web-based platform can be used via a pre-existing central service such as *GitLab.com*, or installed on any kind of network-connected private server. There are two editions of the software: *GitLab EE (Enterprise Edition)*, which has a few extra features for larger organisations, and is the version run on the *GitLab.com* service (for free as of 2015–2017), but cannot be installed on a private server without a license. *GitLab CE (Community Edition)*, in contrast is released with an open license and can be easily installed on any private server.

We opted for the *GitLab CE* private server model for a number of reasons: 1. more control over the storage location of the data; 2. more flexibility as to the number of users who might access the service during the pilot. An instance of *GitLab CE* was set up on a private server managed by the IT department of the German Archaeological Institute that has subsequently been migrated to a server at the University of Cambridge. Once running, a number of different *Git* repositories were set up to separate different activities within the survey project (see figure 2). This included one for files associated with the design of the survey such as template paper and digital *ODK* forms (`panormos/survey-data-design`), one for the data produced from both the *in-field* data recording and *at-base* find analysis (`panormos/survey`), one for the GIS files produced during the project (`panormos/gis-active`), one for storage/backup of photographs (`panormos/photo-archive`) and one for storing the project handbooks and daily logs for the entire team (`panormos/team`). The `team` repository was also used to generate a static information web-site initially exclusively for the use of team members but with the long-term aim of making these documents open to a wider audience as part of the project's methodological transparency.

R

R is a mature, 20± year-old language dedicated to statistical computing (Ihaka & Gentleman 1996). It is actively maintained open and Free Software: as of May 2017, the last release (version 3.3.3 – 'Another Canoe') was only two months old. From the beginning, *R* was designed for statistical modelling and data analysis, but also for outputting well-designed publication-quality 'plots' (i.e. graphs). Perhaps for this reason it is increasingly becoming a dominant analytical language for a number of scientific fields (from biostatistics to finance and sociology). One strength of *R* is its versatility in working in multiple platforms. This makes it much easier to reproduce analytical results with different operating systems on different computers. For our project, this was particularly beneficial as participants were variously using Linux, Mac and Windows and could download and test the code when necessary.

7 Changes made to files between 'commits' (snapshots) can be locally saved but are not tracked, however. All files are effectively treated as drafts until a commit is made.

R is also highly expandable through 'packages', which can be installed via an online repository (most commonly the CRAN library). Over 10000 packages are available; most have been created by the user community to solve general or specific problems. There are also packages that allow *R* to communicate with other software. For example it is possible to call specific algorithms made available by GRASS, SAGA or QGIS in *R*. In the Panormos Project, we relied heavily on those packages designed for spatial/GIS applications, especially `sp` and `raster`.

Supplementary Tables

Table 1. Software and Platforms.

Name	Installation	Weblink	Purpose	FLOSS / Prop.	Integral to Panormos Survey	Initial/Current version Used
AlpineQuest	User	http://www.alpinequest.net/	GIS/mapping app	Prop.	no	
AppEngine	Server	https://cloud.google.com/appengine	Platform to run server software	Prop.	n/a	
ArcGIS	User	http://www.arcgis.com/	GIS/mapping software	Prop.	no	
CVS	User+	http://www.nongnu.org/cvs/	Version control system	FLOSS	no	
dat	User+	http://dat-data.com/	Data version control and sync system	FLOSS	no	
Git	User+	https://git-scm.com/	Distributed version control system	FLOSS	yes	2.1.4/2.9.3
git-annex	User+	https://git-annex.branchable.com/	Manage large files with git	FLOSS	no	
Git LFS	User+	https://git-lfs.github.com/	Manage large files with git	FLOSS	no	
GitLab CE	Server	https://about.gitlab.com/	Web-interface for managing Git repositories	FLOSS	yes	7.12/8.11.5
Jekyll	Server	https://jekyllrb.com/	Static website generator	FLOSS	yes	2.5.3
Mercurial (Hg)	User+	https://www.mercurial-scm.org/	Distributed version control system	FLOSS	no	
QGIS	User	http://www.qgis.org/	GIS/mapping software	FLOSS	yes	2.4/2.16.2
ODKCollect	User	https://opendatakit.org/	Data collection interface	FLOSS	yes	1.4.5/1.4.10
ODKAggregate	Server	https://opendatakit.org/	Data aggregation server (Open Data Kit)	FLOSS	yes	1.4.7
OpenDataKit (ODK)	-	https://opendatakit.org/	See ODKCollect and ODKAggregate			
R	User+	https://www.r-project.org/	Statistical programming language and interpreter	FLOSS	yes	3.2.2/3.3.1
RStudio	User	https://www.rstudio.com/	Development environment for R	FLOSS	yes	0.99/0.99.903
Subversion (SVN)	User+	https://subversion.apache.org/	Version control system	FLOSS	no	

Table 2. Formats and standards.

Name	Weblink	Purpose	Integral to Panormos Survey
ArchaeoML	See Schloen 2001	XML-based text schema for organising archaeological data	no
CSV (Comma-separated values)	https://tools.ietf.org/html/rfc4180	Plain-text format for tabular data	yes
GeoJSON	http://geojson.org/	JSON-based schema for organising geographical data	no
GML (Geographical Markup Language)	https://www.iso.org/obp/ui/	XML-based text schema for organising geographical or spatial data	yes
JSON (JavaScript Object Notation)	http://www.json.org/	Plain-text Javascript-based data notation format	no
Markdown	https://daringfireball.net/projects/markdown/	Lightweight markup language easy to read and write to create rich text using a plain text editor	yes
R Markdown	http://rmarkdown.rstudio.com/	Framework for authoring fully reproducible document with the environment R	yes
XForms	https://www.w3.org/TR/xforms/	XML-based form design schema	yes
XLSForms	http://xlsform.org/	User-friendly XForm design	yes
XML (eXtensible Markup Language)	https://www.w3.org/XML/	A plain-text markup language which can be used to organise data in a semi-structured manner	yes

Table 3. Data and repository services.

Name	Country	Weblink	Purpose	Funding model	Preservation aim
ADS (Archaeology Data Service)	UK	https://archaeologydataservice.ac.uk/	Archaeology-oriented data archive	One-time pay to deposit	Perpetuity (migration)
DANS (Data Archiving and Networked Services)	Netherlands	https://dans.knaw.nl/en	Research data archive	Subsidised	Perpetuity
Figshare	USA	https://figshare.com/	Research data archive	Free	Perpetuity
GitHub.com	USA	https://github.com/	Code-oriented Git repository and code sharing	Free + Subscription	None
GitLab.com	USA	https://gitlab.com/	Code-oriented Git repository and code sharing	Free + Subscription	None
IANUS	Germany	https://www.ianus-fdz.de/	Archaeology-oriented data repository	not finalised	not finalised
OCHRE	USA	https://ochre.uchicago.edu/	Archaeology and text data management platform	Subscription	Perpetuity
OpenContext	USA	https://www.opencontext.org	Archaeology-oriented publishing platform	One-time pay to deposit	Perpetuity
tDAR	USA	https://www.tdar.org/	Archaeology-oriented data repository	Price-per-file one-time-pay	Perpetuity?
Zenodo	Switzerland	https://zenodo.org/	Research data archive	Free	Perpetuity

Table 4. Possible licensing frameworks appropriate for archaeological research data.

Abbrev.	Name	Weblink	Implications for re-use
CC BY-ND	Creative Commons, Attribution, No Derivatives	https://creativecommons.org/licenses/by-nd/4.0/	Author(s) must be credited, data can be copied in its entirety unmodified but no derivative works (i.e. analyses of the data) can be published
CC BY-SA	Creative Commons, Attribution, ShareAlike	https://creativecommons.org/licenses/by-sa/4.0/	Author(s) must be credited, and derivative works (e.g. analyses of the data) must be published/released under the same license (slightly less appropriate for 'data')
CC BY	Creative Commons, Attribution	https://creativecommons.org/licenses/by/4.0/	Author(s) must be credited, copies and derivative works (e.g. further or alternative analyses based on data) are allowed
CC0	Creative Commons Zero, Public Domain	https://creativecommons.org/publicdomain/zero/1.0/	No rights reserved, author need not be credited, copies and derivative works can be made and published under any conditions.