PHILOSOPHICAL TRANSACTIONS A

royalsocietypublishing.org/journal/rsta

Research

Cite this article: Mitchell SN *et al*. 2022 FAIR data pipeline: provenance-driven data management for traceable scientific workflows. *Phil. Trans. R. Soc. A* **380**: 20210300. https://doi.org/10.1098/rsta.2021.0300

Received: 11 October 2021
Accepted: 19 April 2022

One contribution of 18 to a theme issue 'Technical challenges of modelling real-life epidemics and examples of overcoming these'.

Subject Areas:
e-science, software, bioinformatics, computer modelling and simulation

Keywords:
FAIR, provenance, data management, epidemiology, modelling, COVID-19

Author for correspondence:
Richard Reeve
e-mail: richard.reeve@glasgow.ac.uk

Electronic supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.6070465.

THE ROYAL SOCIETY PUBLISHING

FAIR data pipeline: provenance-driven data management for traceable scientific workflows

Sonia Natalie Mitchell[1,2], Andrew Lahiff[6], Nathan Cummings[6], Jonathan Hollocombe[6], Bram Boskamp[7], Ryan Field[3], Dennis Reddyhoff[8], Kristian Zarebski[6], Antony Wilson[9], Bruno Viola[6], Martin Burke[7], Blair Archibald[4], Paul Bessell[10], Richard Blackwell[11], Lisa A. Boden[10], Alys Brett[6], Sam Brett, Ruth Dundas[3], Jessica Enright[2,4], Alejandra N. Gonzalez-Beltran[9], Claire Harris[2,7], Ian Hinder[12], Christopher David Hughes[11], Martin Knight[7], Vino Mano[11], Ciaran McMonagle[2,3], Dominic Mellor[2,5], Sibylle Mohr[1,2], Glenn Marion[2,7], Louise Matthews[1,2], Iain J. McKendrick[2,7], Christopher Mark Pooley[7], Thibaud Porphyre[13], Aaron Reeves[14], Edward Townsend, Robert Turner[8], Jeremy Walton[15] and Richard Reeve[1,2]

[1]Institute of Biodiversity, Animal Health and Comparative Medicine, College of Medical, Veterinary and Life Sciences, University of Glasgow, Glasgow, G12 8QQ, UK
[2]Boyd Orr Centre for Population and Ecosystem Health, University of Glasgow, Glasgow, G12 8QQ, UK
[3]MRC/CSO Social and Public Health Sciences Unit, Institute of Health and Wellbeing, College of Medical, Veterinary and Life Sciences, University of Glasgow, Glasgow, G12 8QQ, UK

[4]School of Computing Science, College of Science and Engineering, University of Glasgow, Glasgow, G12 8QQ, UK
[5]School of Veterinary Medicine, College of Medical, Veterinary and Life Sciences, University of Glasgow, Glasgow G61 1QH, UK
[6]United Kingdom Atomic Energy Authority, Didcot OX14 3DB, UK
[7]Biomathematics and Statistics Scotland (BioSS), James Clerk Maxwell Building, Peter Guthrie Tait Road, The King's Buildings, Edinburgh EH9 3FD, UK
[8]Department of Computer Science, University of Sheffield, Regent Court, Sheffield S1 4DP, UK
[9]Science and Technology Facilities Council, Harwell Campus, Harwell OX11, UK
[10]Roslin Institute, University of Edinburgh, Edinburgh EH8 9YL, UK
[11]Man Group plc, Riverbank House, 2 Swan Lane, London EC4R 3AD, UK
[12]The University of Manchester, Research IT, Manchester M1 3BU, UK
[13]VetAgro Sup, UMR5558 Laboratoire de Biométrie et Biologie Évolutive, Campus vétérinaire de Lyon, Marcy-l'Etoile 69280, France
[14]Scotland's Rural College (SRUC), Peter Wilson Building, The King's Buildings, West Mains Road, Edinburgh EH9 3JG, UK
[15]UK Earth System Model Core Group, Met Office, Exeter EX1 3PB, UK

SNM, 0000-0003-1536-2066; AL, 0000-0002-2785-4116; NC, 0000-0003-4359-6337; JH, 0000-0001-7159-158X; BB, 0000-0002-5446-3209; RF, 0000-0002-4424-9890; DR, 0000-0002-4971-2346; KZ, 0000-0002-6773-1049; AW, 0000-0001-7336-4823; BV, 0000-0001-5406-5860; MB, 0000-0002-5540-9087; BA, 0000-0003-3699-6658; PB, 0000-0002-7901-969X; LAB, 0000-0002-9317-4741; AB, 0000-0002-5789-0572; RD, 0000-0002-3836-4286; JE, 0000-0002-0266-3292; ANG-B, 0000-0003-3499-8262; CH, 0000-0003-0852-2340; IH, 0000-0003-3548-9101; MK, 0000-0002-3812-4397; CM, 0000-0002-1325-2413; DM, 0000-0001-7141-652X; SM, 0000-0002-9089-6327; GM, 0000-0002-0454-9338; LM, 0000-0003-4420-8367; IJM, 0000-0002-7529-0749; CMP, 0000-0002-8779-4477; TP, 0000-0002-8552-9899; AR, 0000-0001-6387-7481; RT, 0000-0002-1353-1404; JW, 0000-0001-7372-178X; RR, 0000-0003-2589-8091

Modern epidemiological analyses to understand and combat the spread of disease depend critically on access to, and use of, data. Rapidly evolving data, such as data streams changing during a disease outbreak, are particularly challenging. Data management is further complicated by data being imprecisely identified when used. Public trust in policy decisions resulting from such analyses is easily damaged and is often low, with cynicism arising where claims of 'following the science' are made without accompanying evidence. Tracing the provenance of such decisions back through open software to primary data would clarify this evidence, enhancing the transparency of the decision-making process. Here, we demonstrate a Findable, Accessible, Interoperable and Reusable (FAIR) data pipeline. Although developed during the COVID-19 pandemic, it allows easy annotation of any data as they are consumed by analyses, or conversely traces the provenance of scientific outputs back through the analytical or modelling source code to primary data. Such a tool provides a mechanism for the public, and fellow scientists, to better assess scientific evidence by inspecting its provenance, while allowing scientists to support policymakers in openly justifying their decisions. We believe that such tools should be promoted for use across all areas of policy-facing research.

This article is part of the theme issue 'Technical challenges of modelling real-life epidemics and examples of overcoming these'.

1. Introduction

Historically, models and analyses used to support advice to government have not been publicly available as public policies are implemented. Typically, some materials would only subsequently become public via traditional publication routes, with the delays that this implies. Technological advances and increasingly influential ideas from open source and reproducible science mean that this approach is no longer tenable. During the current COVID-19 pandemic, many models used by the Scientific Pandemic Influenza Group on Modelling (SPI-M), who advise the United Kingdom Government on human infectious disease threats based on infectious disease modelling and epidemiology, have indeed been made publicly available (e.g. [1–4]). However, even these models still lack the transparent and readily traceable chain of evidence connecting data and assumptions with model outputs that would allow them, and their results, to be readily available and independently assessed. It is also commonly the case that data availability, coverage and quality are extremely variable or, if available, data may not be in a form that can be easily used without curation or transformation, further analysis and a detailed description of the data. The ephemeral nature of some data sources and the rapid evolution of datasets used during an emergency, combined with the sparsity or absence of metadata describing datasets, all compound the problem of assessing evidence. In this article, we examine the challenges around data coverage, quality and access with a particular focus on the issues and demands highlighted by the COVID-19 pandemic and outline the properties of a data pipeline designed to provide an infrastructure to address the demands of disease modelling for outbreak control policy-making, now and in the future.

The modelling work during the COVID-19 pandemic to generate estimates of key parameters and make predictions of likely outbreak trajectories has required multiple epidemiological models, operating at different scales and with varying levels of epidemiological detail (e.g. [5–8]). This raises issues around the scale and resolution of the data used, and the extent to which the data are processed or abstracted before use. Moreover, the existence of multiple models, drawing on the same pool of available datasets, but in different ways, exposes a key point: that data, models and results are all research objects that require management.

From a practical perspective, the pandemic has made it clear that modellers must make the basis of their advice both transparent and accessible. Following the path from basic science to policy-friendly interpretation, via choice of parameter values, model structure, model assumptions, code implementation and generation of outputs, is complex even for specialists. Version control tools like `git` combined with online repositories such as *GitHub* have hugely enhanced sharing and collaboration on code, and managed repositories such as *Zenodo* now exist for Open Science. However, this is just a small subset of what is required from a usable platform to support open and transparent epidemiological modelling, given the requirements that it be consistent in use across the range of likely applications, sufficiently unobtrusive that it is feasible for modellers to adopt it, and accommodating the necessary diversity of data sources. Digital Research Infrastructure can now support transparent linking of the steps along this pathway.

We have developed a pipeline that provides an open and publicly accessible 'chain of trust' to transparently connect primary data with research outputs via open source, publicly accessible analysis and modelling code. This pipeline provides a route by which scientific hypotheses and study results combined with other sources of societal data (e.g. epidemiological, demographic, geographic and health service use) can contribute, through intermediate analyses, to publicly available and openly tested models while enabling generation of outputs whose dependencies can be fully interrogated.

In developing this pipeline, we provide an Open Science solution that addresses long-standing and critical problems in public health and livestock disease control. For example, Keeling [9], in a review of the modelling effort during the UK Foot and Mouth disease outbreak of 2001, identified a number of sources of conflict associated with the use of mathematical modelling in emergency veterinary public health. Keeling argued that tension between the veterinary and modelling sciences arose at least partially because of a disjunction of experience at different scales:

that veterinary expertise was likely to be more accurate and effective at a local scale, whereas models were most effective in integrating the risks associated with multiple, larger scale events. It therefore becomes difficult to seek consensus across these groups, due to the confounding of perspective with professional expertise. A more general point was made by Matthews *et al.* [10], who argued that, as the spatial scale on which decisions and interventions are required increases, the threshold at which it is worthwhile to intervene, when measured in terms of the estimated risk of infection per premises, will tend to decrease. In this case also, we can note a problematic confounding of perspective (local, regional, national or supra-national) with the properties of any models operating at these different scales. Unfortunately, it is all too easy to see how these systemic issues could translate into a lack of confidence in the results of a quantitative analysis, or even into a dismissal of modelling results where these conflict with the 'common sense' of an influential grouping.

One way to overcome these potential problems is to seek to maximize the transparency of the modelling process, opening to general scrutiny the logic which has given rise to potentially contentious results, clearly reporting analytical assumptions and the provenance of the data resources, which have underpinned the analysis. To build scientific, political and public consensus, at a minimum, it is desirable to avoid mistakes arising from poor management or weak understanding of the data resources used. It is also important to avoid propagating any errors which do arise and essential to have tools to find any such issues. The importance of maintaining transparency and supporting better management of provenance of data, data products and modelling outputs is underlined by the official statement made by the UK Office for Statistics Regulation in November 2020 [11]. Three key objectives were specified in respect of governmental use of data to support COVID-19 decision-making: namely that (i) where data are referenced publicly, that the data or at minimum the provenance of the data be published; (ii) where models are referenced publicly, that model outputs, methodologies and assumptions also be published; and (iii) where decisions are justified by reference to data, that this be made publicly available. These objectives are supported by the functionalities of the Findable, Accessible, Interoperable and Reusable (FAIR) data pipeline described below.

This article initially describes the issues we are trying to address and the existing tools that partially address them. It then follows the conceptual development and implementation of the open source pipeline to connect baseline assumptions and data to epidemiological models and their outputs. Using exemplary epidemiological models, we demonstrate how model runs are associated with a specific, cumulative chain of dependencies, supporting the critical examination of assumptions. If errors (or issues) are identified in primary datasets, analyses or modelling code, downstream model outputs can be automatically and transparently invalidated. The finalized infrastructure aims to provide an ecosystem for the epidemiological and wider scientific communities within which data, models and results can be managed in a transparent and publicly accessible way.

(a) FAIR research objects, provenance and data cataloguing

The FAIR data principles [12] were proposed as guidelines to apply to data, making them findable, accessible, interoperable and reusable so that both researchers and machines are able to find, access and (re)use data. The principles have been widely adopted and subsequently have also been applied to other research objects such as software [13], workflows [14], machine learning algorithms and executable notebooks; and guidance for 'FAIRification' of data has been developed [15,16].

The main data descriptors required to achieve FAIR data according to the principles are as follows: (i) persistent identifiers, such as digital object identifiers (DOIs, [17]) for data, metadata and software, Open Researcher and Contributor IDs (ORCIDs [18]) for people, and (ii) standardized ways of recording metadata.

As regards metadata standards, the open source pipeline requires a data registry that is aligned to existing formats and vocabularies. This will enable interoperability with other systems and

support integrating and exchanging data in a straightforward way. To reduce the ambiguity in diverse data representations, we rely on common formats (such as JSON-LD [19]), and common terminologies or FAIR vocabularies [20] that provide clear definitions and persistent identifiers for the terms. For example, we use the *provenance vocabulary* (PROV-O) to faithfully represent the entities, activities and people involved in producing a research output.

2. Requirements analysis

The Scottish COVID-19 Response Consortium (SCRC) [21] was created as part of the Rapid Assistance in Modelling the Pandemic (RAMP) initiative, coordinated by the Royal Society [22]. During 2020, the consortium comprised over 150 volunteers from multiple universities, research centres and industrial partners across the UK. During the design phase, we carried out a requirements analysis involving modellers, epidemiologists, data and policy experts and software engineers from across the consortium to determine how such a pipeline would be used, document these use cases and extract from them the most important requirements. A search was made for existing technologies that might satisfy these requirements, but it yielded no results (see §3 for details), and a prototype pipeline was built to investigate the pros and cons of different approaches. In the light of this exercise, the current pipeline was then designed and built to meet the specifications.

(a) Use cases

From the identified use cases for the FAIR data pipeline, ultimately 17 were taken forward and written up in detail. These use cases generated a variety of requirements, such as being able to query the pipeline to easily establish whether a dataset is already registered; being able to run analyses using the pipeline inside a Trusted Research Environment (TRE); being able to raise issues with data or software at run time or retrospectively to track the quality of these entities; and to inspect an output generated by a third party and identify whether any issues have ever been raised with any component in its provenance. An important criterion established in the use cases was that the software be easily approachable with low technical barriers of entry, thus making the pipeline accessible to end users.

(b) Requirements

In the context of our objectives for a pipeline for epidemiological models, and mindful of the requirements specified (above) by the Office for Statistics Regulation, we require:

(i) A FAIR representation of the research objects involved, such as datasets and software, and the ability to trace updates to them, identify specific versions being used in analyses, track their provenance and integrate all the information necessary to understand how results are produced. This should include the ability to manually add provenance to the system where, for instance, a policy report contains figures, or simply results, generated by the pipeline;

(ii) The ability to work seamlessly with data that is not publicly available, and indeed to be able to work fully offline, for instance in situations utilizing sensitive data inside TREs such as National Health Service (NHS) Safe Havens. In these situations, it is still necessary to be able to uphold the FAIR principles and to allow for the evaluation of the provenance of research outputs by providing public access to metadata. However, the data must be provably isolated from this process to comply with access requirements;

(iii) Interoperability with existing standards. No approach to solving these problems will be successful if it invents a series of new 'standards', when a plethora of existing standards already exist. This is because so much data already exist in repositories that already comply with these standards, and users must be able to easily pull these datasets into

the pipeline. This does not presuppose that existing standards have to be used to define internal formats, but at least there has to be the ability to interoperate through seamless import and/or export (e.g. DOIs, W3C-PROV [23] and W3C-DCAT [24]); and

(iv) That the system is not disruptive for end users, providing clearly identifiable short-term benefits to epidemiologists, and the Research Software Engineers working with them, to encourage uptake while causing as little friction as possible within their existing workflows.

While reproducibility of results is desirable and may be possible where the data used are publicly available, it is not a core requirement since our concern with simplicity and user-friendliness (to minimize barriers of entry to modellers) can be in conflict with, for instance, containerization approaches (containers package up code and all of its dependencies to allow quick and reliable transfer of analyses from one computing environment to another) that allow full reproducibility in all circumstances.

A number of other requirements were identified for individual use cases, but since they are mostly very specific to individual problems, and can be solved by (for instance) simply refining interfaces to help end users use existing functionality, they are not listed here.

3. Related work

Our FAIR Data Pipeline must combine components for executing modelling workflows as well as recording the information and provenance of the research objects. This combination enables traceability of the modelling results.

In this section, we review related systems that provide similar functionality. We categorize them as (i) those providing databases or online data repositories, often focused primarily on data, (ii) those providing ways of recording workflows, usually focused primarily on reproducibility, (iii) those recording the provenance of research objects by tracing code as it executes, and (iv) those providing or relying on version control systems, often combining software and data repositories. However, note that there is a significant overlap between these categories, so we have endeavoured to place the tools in the most appropriate category.

(a) Online data repositories and databases

The ability to store data was not a core requirement of our data pipeline because of the range of storage solutions already available. Indeed, the specific storage mechanisms at individual sites, such as inside NHS Safe Havens, are well established and unlikely to change. Nonetheless, existing data storage solutions offer a partial solution to the issues that we are trying to address.

For instance, Zenodo [25], and other online repositories such as figshare [26], provide persistent identifiers (DOIs) for any form of data, and also record standard associated metadata. By contrast, FAIRDOMHub [27], and the underlying FAIRDOM-SEEK [28] software, provide a FAIR data and model management service specifically for Systems Biology, with metadata tailored specifically to this community.

Splitgraph [29], on the other hand, is a PostgreSQL-based tool for building, versioning and querying reproducible datasets. It provides a public data store, while allowing provenance tracking of datasets that are created within Splitgraph. Dolt [30] (together with the associated online repository, DoltHub [31]) provides similar SQL functionality, with commercial options for private data storage, but does not generate provenance. Fully commercial data stores, such as data.world [32], also offer similar capabilities.

All of these data stores offer some of the functionality required by the pipeline, but all would require significant changes to users' workflows and none can be used offline. Nonetheless, the ability to interoperate with existing data stores like these, particularly public ones such as Zenodo, would be very valuable for accessing existing published datasets.

(b) Reproducible workflows

As discussed in §2b, computational reproducibility was not seen as a critical requirement for the pipeline during the development of the use cases [33]. However, a key requirement was the ability to trace exactly what code was run on which datasets to ensure accurate provenance recording. Traceability can be seen as an indirect element of a reproducible workflow; therefore, there are overlaps with tools that already exist for these purposes.

Highly developed tools exist for managing and scheduling workflows [34] that generally guarantee reproducibility, such as Galaxy [35] for scientific workflows, Apache Airflow [36] for more general data pipelines, and before that, Kepler [37]. Such tools are generally designed for complex and/or regular workflows, not the bespoke and one-off analyses that are more frequently produced in academia and which are detailed in our use cases.

On the other hand, Kaggle [38] provides an online repository containing a wide variety of publicly available datasets and a cloud-based Jupyter Notebook [39] environment for reproducibly analysing these datasets. However, the Jupyter-based workflow does strongly constrain what it is possible to do with the system, and the utility of cloud-based systems may also be undermined by legal requirements (e.g. General Data Protection Regulation (GDPR) [40]), which place requirements on data processors not to export data outside specific jurisdictions. Quilt [41] is an open source data hub that allows analyses to be run in (much more general) Docker containers [42]. Quilt also provides a commercial product, QuiltData [43], for managing private data. Neither of these explicitly allow provenance to be extracted, but the information is indirectly available through the contents of the notebooks and containers. A different issue with commercial platforms for data management and reproducibility is exemplified by FloydHub [44], which provided similar functionality to QuiltData, but the company that provided this service has ceased trading during the writing of this article.

At least two tools were created during the pandemic specifically for reproducible analyses of COVID-19 data. Covid Model-Runner [45] was created in the early months of the pandemic to automate the epidemiological analysis of COVID-19 datasets that predicted future outcomes under different control scenarios. It used Docker to containerize the analyses and enforced a standard input and output schema for the models to ensure easy comparability. However, despite the excellent work that went into it, the tool was not widely adopted, perhaps due to its narrow focus, combined with the complexity of adapting code to use it. The lesson from this work may simply be that such tools must be adaptable to the workflows of the end user and cannot assume that the converse will apply, unless they provide a critical (to the epidemiologist) service not otherwise available. The second platform that has been developed during the pandemic has had a very different trajectory, providing as it does just such a critical service—OpenSAFELY [46,47] was developed to allow open and reproducible analysis of sensitive NHS patient data. It holds electronic health record data for 40% of the population of England, and nine papers have already been published using the platform, providing important results about the disease and the efficacy of treatments. OpenSAFELY is designed for the analysis to be fully open, with all activity publicly logged. It can create its own TRE to operate appropriately on sensitive data or can be deployed on top of an existing TRE as a privacy-enhancing layer. It has many desirable features that enhance privacy such as the lack of access by users to raw data even when analyses are being conducted. However, it is expressly designed to operate solely inside TREs, and as such it puts strong constraints on how users interact with it, making it much less suitable for use on non-sensitive data.

These tools for reproducible workflow management provide useful functionality, but many require even more substantial changes to user workflows than the data management systems above. For specific tasks working with sensitive data, OpenSAFELY provides totally new levels of privacy protection, but outside that narrow focus, we believe that the constraints such tools impose are too heavy for our uses.

(c) Provenance tools

Data provenance, or pedigree or lineage, is about documenting the processes that produce the data in its current form. Much of the early work capturing provenance for scientific workflows is summarized in a conference demonstration [48] and a review [34] in 2008. Since then, a standard has emerged for documenting data provenance—the W3C Provenance Ontology [23]—and there are multiple tools that support recording provenance in different contexts. Here, as well as tools that capture provenance through workflow management (e.g. [36,37], described earlier), we classify provenance tools into three further groups—tools that work at the level of the operating system, tools that generate provenances for specific languages and tools that manage provenance data for multiple languages.

Camflow [49] is an example of an operating system-level tool, capturing 'whole system provenance' [50]. It captures the relationships between Linux operating system kernel objects (such as files and threads) during execution and stores these such that they can be represented as a directed acyclic graph. This is a very powerful tool, but captures so much information it is hard to use for a relatively 'simple' modelling task. RDataTracker (and rdtLite and rdt) [51] are tools to collect provenance information from code executed in R, outputting to PROV JSON format. This can be visualized as a data derivation graph showing how data and computation led to an result. It requires no alteration to the underlying R code, providing a very low barrier of entry, but is inappropriate for our use, both because it slows down the code significantly in order to trace it, and it only works on R code, strongly constraining the provenance that can be captured. A recordr [52] is another tool working only in R to record provenance of analyses run in R without altering the source code. Unlike RDataTracker, it does not significantly slow down the code, but is limited by only instrumenting a very limited number of function calls that read and write files. However, it works directly with the DataONE earth and environment online repository (https://www.dataone.org/), allowing metadata to be recorded for some of the data being used by scripts being traced. Finally, recipy [53] is a provenance tracking tool for Python. Again it requires only minimal changes to the code to be tracked, but is limited, in both that it only tracks code in Python, and that it just traces file paths, and does not retain the data or store any metadata on the data being read and written. The Core Provenance Library (CPL) [54] provides an interface to a relational database that is used to store provenance information. It is written in C/C++ and provides interfaces in R, Python and Java as well. Developers integrate these interfaces into their code and actively use them to store provenance information. However, the provenance APIs provided are very low level, not automatically capturing provenance as code is run, but rather providing the ability to manually create, look up and link objects in the database, making the barrier of entry just too high for most users.

These provenance tools are all very valuable in their own right, and CPL is the closest of the tools to our requirements, but we believe that the barrier of entry for such a tool is just too high for our users, while the other, simpler approaches are too constrained in what they can track, and conversely, the whole operating system approach is just too complex for our needs.

(d) Version control

Source code for software has been curated through the use of version control software such as `git` [55] for many decades, and there is widespread adoption throughout the epidemiological modelling community (e.g. [1,2]). However, version control for data is less well developed, not least because, unlike source code, data often involve files that traditional version control cannot easily handle at present, whether due to limitations in memory, time or disk space. To allow large and binary data to be more easily managed inside version control, additional functionality has been added to `git` through `git annex` [56], which uses special remote stores including cloud object storage to manage data outside the git repository. However, this is not trivial to use, and so other platforms have been built on top of `git annex` to allow end users to access this functionality more easily.

Some of the data management tools described in §3a provide forms of version control. These are often quite primitive in the sense that they do not identify specific changes to datasets, but only that the dataset has changed. Qri [57] is an open source tool for data management, versioning and sharing. Versioning happens at the individual dataset level, with commits containing high-level metadata on the data as well as more detailed information on its structure. Pachyderm [58] is a platform for reproducible science with versioned data repositories. It too provides provenance at the level of data repositories that are used and produced by analyses, and reproducibility through Docker images and Kubernetes [59] for container management. However, it is a commercial product that has job limits on the free community edition, and it carries only limited metadata on the data it tracks. Data Version Control (DVC) [60] (built on `git annex`), and DAGSHub [61], which uses DVC for data management, `git` for software version control and MLflow [62] for reproducibility, provides similar functionality to Pachyderm including provenance tracking and is designed especially around machine learning workflows. DataLad [63] is also built on top of `git annex` and provides similar functionality to DVC and Pachyderm, but has a broader scope.

We believe that the tools described earlier, especially the open source DVC/DAGSHub and DataLad platforms, come the closest of all of the identified existing products to satisfying our requirements. They provide provenance information (though it is not clear if manual additions can be made), they can handle private and public data, and they are relatively unintrusive in how users interact with them. However, they still fall short in several significant respects. Critically, it is not clear how to separate the metadata from the data itself, since `git` and `git annex` appear to manage both 'under the hood', and so it is not clear how to ensure public traceability once part of the provenance of a model output is marked as belonging to a private data store. The metadata they provide is also not clearly interoperable with formal standards; although the provenance information can almost certainly be standardized, metadata standards go far beyond that. Users and organizations interacting with the system should be traceable using persistent identifiers such as ORCIDs, and the data management system should track persistent identifiers for datasets such as DOIs (see §4a for further details).

All of the tools described earlier are valuable, and some are widely adopted, with many obvious applications. Many positive lessons were identified from them in developing our pipeline, but they are all missing core features that make them unsuitable for our purposes.

4. Overview of FAIR data pipeline

(a) Standards and interoperability

Our solution satisfies the requirements for both syntactic (referring to the formats used for the representation of the data) and semantic (referring to the vocabularies used for the representation of the data) interoperability of metadata by relying on the following standards and technologies:

JSON-LD: A standard format that extends JavaScript Object Notation (JSON) for Linking Data to allow automated navigation from one piece of Linked Data through embedded links to other pieces of Linked Data across the web; we use a JSON-LD representation for each of the entities and for the provenance report [19].

W3C-PROV: A vocabulary for the provision of information about people and activities, such as running code, that are involved in producing a data product in the pipeline, and representing the data provenance [23].

W3C-DCAT: A vocabulary for data cataloguing that enables us to describe datasets and facilitates the consumption and aggregation of metadata from multiple catalogues [24].

DOI, ORCID and ROR: Persistent identifiers for uniquely identifying digital resources, people and organizations (respectively), and providing associated metadata.

We will continue to work on expanding the metadata representation, provide export of data in the relevant formats and plan to package the data products using the **RO-Crate** approach. RO-Crates [64], or Research Object Crates, are a lightweight, JSON-LD-based approach to packaging research data with their metadata, providing a standard import and export format.

(b) Flexible, easy to use and secure

The pipeline must fit easily into existing workflows for data access, processing, modelling and analysis, and to support those likely to be required in a crisis situation, while ideally reducing, and certainly not adding to, the workload of users. Such workflows might involve exploratory work on a scientist's local computer, the need for code to run on HPC nodes without direct runtime internet access, or working within data safe havens and similar restricted environments.

Many of the platforms described earlier rely on cloud-based solutions or heavily constrain what workflows are possible, providing a barrier of entry sufficiently high that they will never be adopted by the target audience. Satisfying the requirements for both simplicity and working with sensitive data led instead to our adoption of a distributed architecture with local data registries that can operate fully autonomously on a laptop or in a secure environment, and which can wrap any existing workflow with minimal changes to ease adoption of the pipeline. Since these local registries contain only metadata, which is not disclosive, they are able to synchronize with remote registries to satisfy requirements for public accessibility of provenance information even for sensitive analyses.

(c) Trust and quality

During the RAMP period, SCRC proposed a model evaluation framework for open epidemiology that would ensure information about provenance, quality and robustness of modelling results would be available alongside any advice or reports that may be used in decision-making. This would cover the key elements contributing to the validity of modelling results including the quality of the underlying science, confidence in the correctness of the software implementation and the reproducibility of outputs from it, the existence and results from validation of models and inference procedures, and the quality and provenance of data, combining it into an overall assessment of output policy-readiness.

The FAIR data pipeline enables such evaluation reports to be attached to objects in the provenance chain. SCRC's lead Research Software Engineers developed a software checklist [65] intended to be completed by software developer(s) and updated for each release of their software. It can then be associated with the release through a dedicated table in the data registry. This checklist has been completed for key components of the FAIR data pipeline with copies stored as `software-checklist.md` in the root of each GitHub repository alongside the software source code.

(d) Overview of components

The FAIR Data Pipeline software suite (table 1) consists of (i) the **data registry**, a Django database-backed web application holding metadata and providing REST APIs; (ii) `fair`, a command line tool that can both synchronize metadata (and optionally data) between the execution platform and a remote data registry and is used to start experimental runs directly; and (iii) a set of language-native programming interface packages, which can be added as dependencies of modelling software to enable it to read and write data from the pipeline. See table 1 for package details, and figure 1 for how metadata and data flows through the pipeline. Provenance is tracked automatically by launching analyses through a command-line tool, and tracking files as they are read and written by minimal editing of the modelling code to wrap the read and write calls.

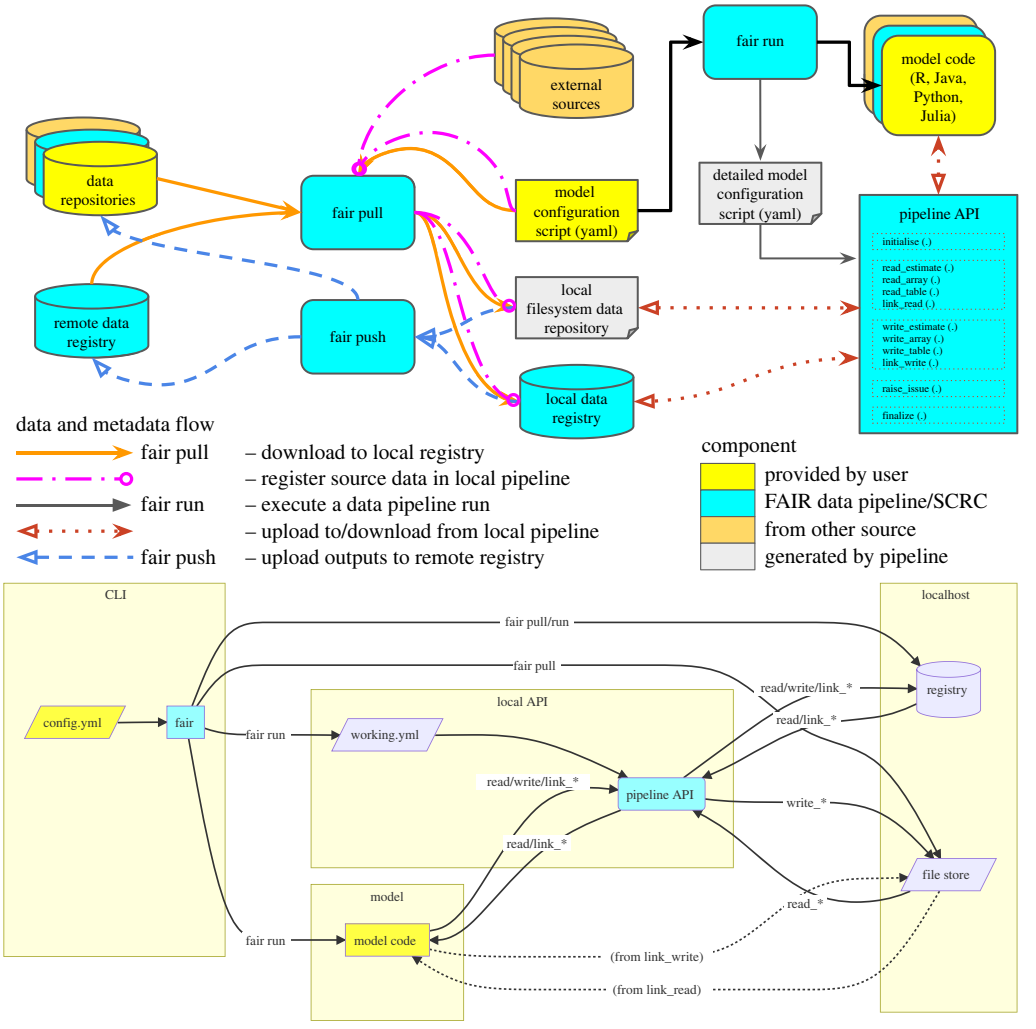

Figure 1. Design of the FAIR Data Pipeline. Upper panel: control flow for the whole pipeline showing local and remote registries, and actions that can be taken to populate and synchronize them. Lower panel: control flow for the local pipeline, showing actions that can occur during a model run. (Online version in colour.)

5. FAIR data registry

The *data registry* is a dynamically updated database that stores metadata associated with the diverse types of data utilized and generated within a typical epidemiological modelling workflow. In this context, we refer to data in the broad sense as any files, including scripts, code, figures and individual parameters. The system design anticipates a variety of different research objects being present in the modelling workflow, whose interactions give rise to new objects that are automatically entered into the registry. The registry was designed to hold the following research objects:

datasets either sourced from a data provider, such as a government department or agency, as open data accessed from another researcher, or generated as part of a registered workflow;

version-controlled processing scripts typically written by the team using the registry, to

Table 1. FAIR Data Pipeline software: all packages are available under open source licenses and are developed as public repositories within the FAIRDataPipeline GitHub organization [66], and where appropriate, released through language-specific package registries.

repository name	language	registry	package name
data-registry [67]	Python (Django)	n.a.	n.a.
FAIR-CLI [68]	Python	PyPI	fair-cli
rDataPipeline [69]	R	CRAN[a]	rDataPipeline
DataPipeline.jl [70]	Julia	General	DataPipeline
javaDataPipeline [71]	Java	Maven	org.fairdatapipeline
pyDataPipeline [72]	Python	PyPI	data-pipeline-api
cppDataPipeline [73]	C++	n.a.	libfdpapi

[a]Package to be added.

- restructure datasets, including activities such as identifying anomalies and raising issues, data cleaning, selecting subsets and linking records across datasets,
- carry out curation activities on datasets,
- carry out specific analysis, producing output that is either going to form part of another output in its own right, or numerical quantities (typically a vector of parameters) that will be used downstream in the workflow;

version-controlled analysis scripts typically written by registry users;

analysis outputs generated by the application of an analysis script to a dataset;

model parameters either extracted from an analysis output, or entered into the registry as a quantity derived from the work of the wider scientific community, such as a parameter cited in a paper or a report from another modelling group;

version-controlled modelling code typically written by registry users for use in generating specific model outputs;

mathematical model output generated by use of a version of a mathematical model codebase, probably making use of multiple datasets and model parameters;

reports generated by pipelining model and analysis outputs into a pre-specified report format, or generated manually (e.g. in Word) using such outputs.

The metadata associated with these research objects can be either intrinsic or extrinsic. Intrinsic metadata includes fields that contain information relating to provenance. Data objects that enter the registry from outside the data pipeline have information uploaded to detail the source of the material, preferably including a persistent identifier such as a DOI as a commonly recognized, persistent method for machine actionable and globally unique identification. Data objects that enter the registry having been produced by researchers working within the data pipeline will have metadata associated with them automatically detailing author and versioning. Such outputs will also trace their history through pipeline interactions to uniquely identify the provenance of the new objects. The operation of the data pipeline will not lead to any revision of the intrinsic metadata associated with a data object; these may need to be updated if further information about provenance becomes available, but this reflects a change in the user's knowledge, not in the actual nature of the data. A key property of the registry is that changes in the metadata associated with a data object will propagate to the provenance metadata associated with offspring data objects. This is useful in maintaining consistently valid, high-quality intrinsic metadata across the entire population of research objects in the registry, but it is even more important when considering extrinsic metadata.

By contrast, extrinsic metadata can be updated over time. The data registry includes two key elements of extrinsic metadata: one ('QualityControlled') is an assessment of the quality, or

fitness-for-purpose, of the research object, whether data or code. The other ('Issue') can be used specifically for raising concerns about problems identified in the data or code, either at the point of data upload or generation or through later analysis. The status of the dataset is dynamically propagated through the registry and hence is visible in the provenance of outputs generated using these datasets. In a similar way, datasets that have been simply superseded can also be flagged, and reports and other outputs based on them can then be identified. If such a report is used as the basis of (say) a briefing to government policymakers, it will be important to make it clear where it is indeed based on outdated information. The key outcome is that all the information required to contextualize the work is available in the provenance metadata. It is instructive to compare this situation with the more extreme case where a dataset is (say) discovered to have contained erroneous data for a period of time. All versions of the dataset subsequently identified as erroneous can be flagged as invalid. All outputs derived from these datasets will have this invalidation in their provenance; reports can be withdrawn, and there should be no risk of any work subsequently using these invalid data without this being apparent in the provenance metadata. In this way, the registry is the primary source of information to derive retrospective provenance, i.e. a detailed log of the execution of the computational task, including user-defined provenance in the form of annotations [34].

The full schema of the registry database, including other metadata such as information about software releases and DOIs for published data, is available in electronic supplementary material, figure 1.

6. Examples

We have selected three examples to demonstrate features of the pipeline. These are not intended to push the limits of the software framework, but to provide simple to understand examples that can be replicated by the reader to get a feel for the complexity of using the FAIR Data Pipeline. The first is the reproduction in R of a simple Susceptible-Exposed-Infected-Recovered-Susceptible (SEIRS) epidemiological model used by Bjørnstad et al. [74] to demonstrate disease dynamics. In the second example, we reimplement this model in all four native languages of the data pipeline (R, Java, Julia and Python) and cross-validate the results. Finally, we show a more complex, but nonetheless very simplified, time-varying model of COVID-19 dynamics with parameters extracted from English epidemiological data from the pandemic, and pull these into the pipeline and run a deterministic simulation of the pandemic using those inferred parameters.

(a) SEIRS model

To demonstrate a simple epidemiological model being run through the pipeline, we take the example of the SEIRS model used by Bjornstat *et al.* [74] and reproduce the results that lead to fig. 1*b* in that paper. The full code is available online in an R package [75], which provides instructions for how to run it in the `README.md` file on GitHub. A vignette is also provided [76] showing the fully worked example with results. `fair pull` is used (figure 1) to populate the local registry with the parameters from the source manuscript (using a `register` block in the configuration file to ensure the data are in the pipeline). `fair run` is then used to execute the R script (the R code is executed from the root of the git repository via the `script` block of the configuration file). The R code itself is only very lightly edited from an equivalent non-pipeline version, with the addition of `initialise()` and `finalise()` steps to start and stop the pipeline monitoring, and the replacement of hard-coded file paths with calls to `link_read()` and `link_write()` with references to labels given in the configuration file (by default the unique names under which they are stored in the registry). These functions simply return appropriate paths, so can be directly used in place of filenames in any normal R code. The output of this simulation is shown in figure 2 along with the provenance of the output figure.

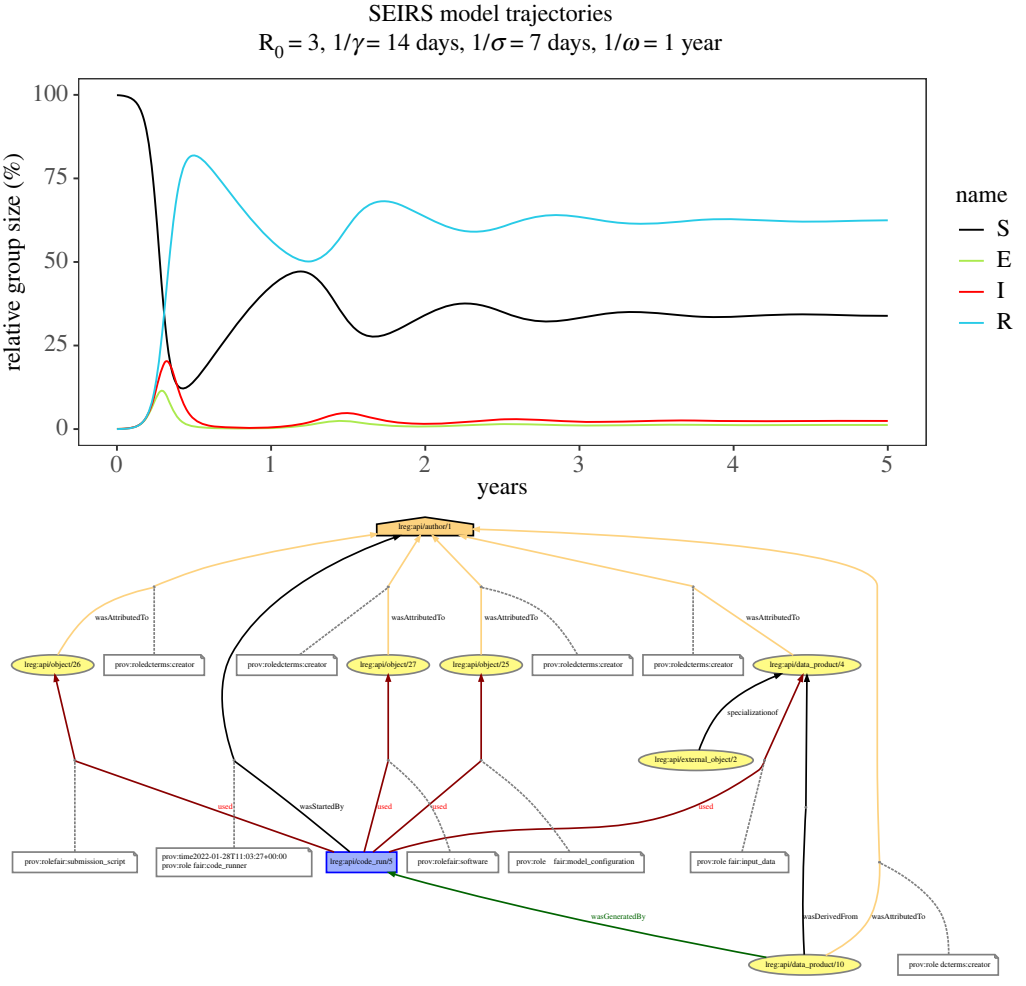

Figure 2. Running an SEIRS model in R through the data pipeline. Upper panel: output of the model, showing SEIRS dynamics matching Bjorstadt *et al.* [74]. Lower panel: provenance of the model output, tracing the SEIRS plot back to parameter inputs. Note that provenances are ordinarily provided in standard PROV-O format, hyperlinked to the research objects being traced and with additional metadata, but they are simplified here for display purposes. (Online version in colour.)

(b) Model comparison

The SEIRS model described earlier was implemented in all four of the native languages of the data pipeline, and the Java [77], Python [72] and Julia [70] implementations can be found online. We ran all four models through the pipeline together and then wrote a small cross-validation script to compare the models. The comparison is shown in figure 3. Critically, the models disagree due to a difference in time steps used and the length of a year in the implementations (365 versus 365.25 days). Examining the provenance of the figure shows that this was automatically identified by the cross-validation code, which then added an issue to the Java model output (the issue can be seen in the registry interface in the electronic supplementary material, figure 2). This issue can then be traced through the provenance to anything that uses this data product. This ability to trace problems with downstream research objects (in this case figure 3) resulting from (even retrospectively) identified problems with upstream data is a core strength of the pipeline.

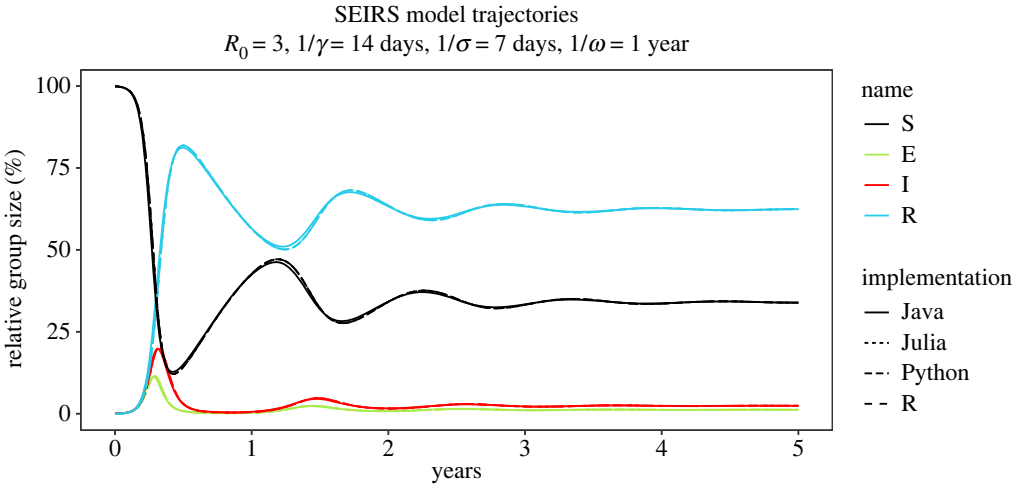

Figure 3. A comparison of all four language implementations through the data pipeline. This figure shows the plots as almost superimposed. The reason for the slight discrepancy was automatically identified during the cross-validation, and in consequence, the provenance of the figure highlights the issue raised. (Online version in colour.)

(c) COVID-19 model

Epidemiological models such as the one described earlier are critically dependent on the values of parameters; these are typically difficult to estimate and subject to variable levels of uncertainty. It is therefore important to be able to trace model outputs back to the parameterization(s) used to generate them. We illustrate this point in a case study in which the parameterization was determined by fitting the epidemiological model shown in figure 4 to COVID-19 epidemic data from England up to mid-2021. These include the static parameters (figure 4 (upper panel)) and the time-varying external force of infection e_t and basic reproduction number R_t that account for the impact of pandemic response (e.g. lockdowns and travel restrictions) on the outbreak dynamics. The model runs shown in the lower panel of figure 4 are generated by using the FAIR data pipeline to link the deterministic, R-based implementation of the model to the parameter values generated via Bayesian inference (figure 4 caption). The full code is provided in the same git repository as the first example [75], and a vignette is also provided showing a fully worked example of it being run in the pipeline [78]. Where the Bayesian inference is itself carried out within the data pipeline, the provenance information will itself chain back to incorporate both the stochastic model and the primary datasets.

7. Discussion

During the COVID-19 pandemic, media outlets have channelled highly charged and politically polarized arguments about the trustworthiness of scientific advice for government policy and also of the advisors themselves, as well as debating the extent to which governments are, in any event, following such advice. While some such controversies are inevitable, as a scientific community endeavouring to provide the best advice we can to policymakers, we are at a disadvantage if the detail of our results is hidden and if the evidence chain that connects our advice to the data and models that underpin it is not just unavailable to the public, but in fact largely non-existent. This situation has arisen although standards have been available for many years, promoting openness and reuse of data and metadata, in particular through the FAIR principles for scientific data management [12].

The FAIR Data Pipeline was SCRC's response to this aspect of pandemic response. Querying epidemiologists involved in both human and animal disease modelling, we could identify no tools being used that satisfied either FAIR principles or which publicly presented the provenance

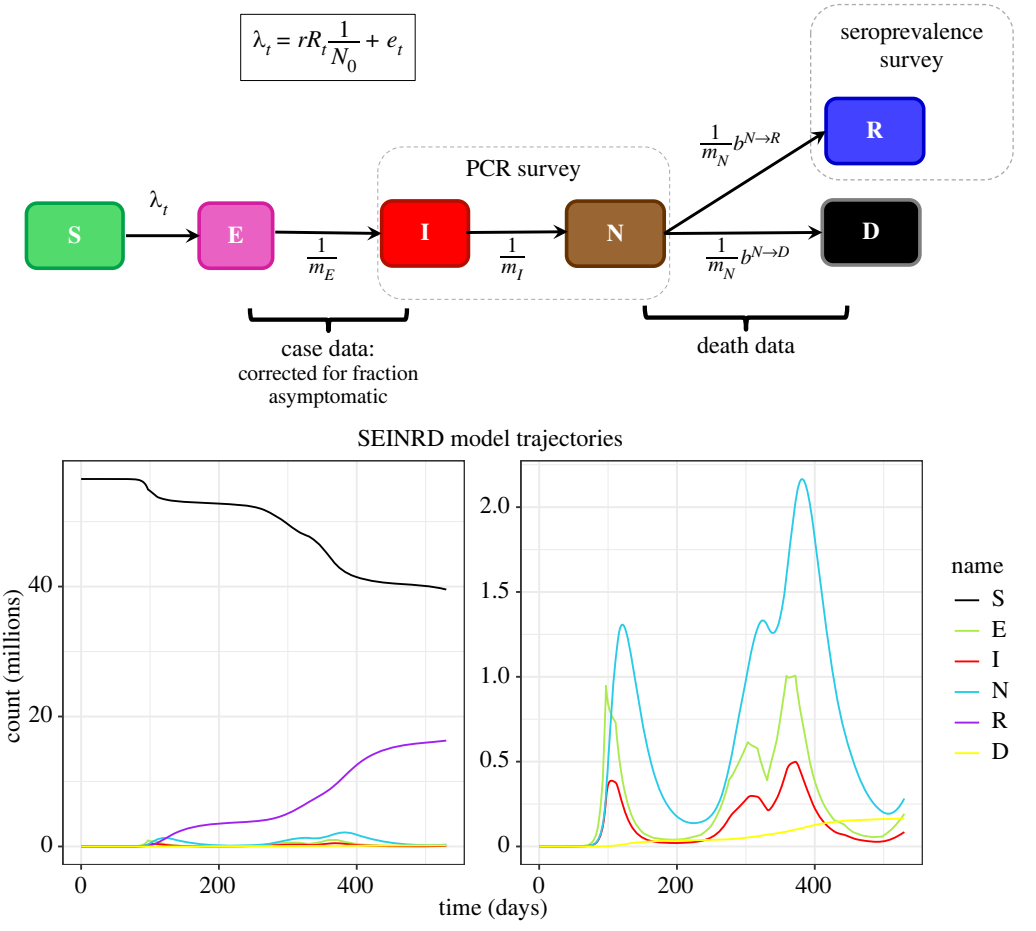

Figure 4. Epidemiological model fitted to 2020-2021 COVID-19 data from England. Upper panel: this shows a homogeneous model (i.e. with no age or spatial structure) with compartments representing, for each point in time, the number of individuals who are susceptible S, infected with COVID-19 but not infectious E, infectious I, isolating/non-infectious N, recovered R, and those who have died D. The figure shows the per-capita rates of transition for each allowed transition between states. The parameters m_E, m_I, m_N represent the average time spent in the states E, I and N, respectively, while $b^{N\to D}$ and $b^{N\to R} = 1 - b^{N\to D}$ are the probabilities of death and recovery. The force of infection λ_t is dependent on: a per-capita external force of infection e_t; and a frequency-dependent term rR_tI/N_0 that represents the rate of disease transmission in terms of the average time spent in an infectious state (here $r = 1/m_I$) and the real-time reproduction rate R_t. Both R_t and e_t vary with time and are driving variables for the model. As indicated, case and death rate data inform the transitions shown, whereas PCR and seroprevalence data from the Coronavirus Infection Survey [79] inform on the numbers of individuals in the compartments $I + N$ and R, respectively. With the exceptions of $m_E = 4$ days and $m_I = 4$ days, all model parameters were determined via Bayesian inference using the methodology described in [80] applied to a stochastic version of the model described here. Note that, although the latent and infectious periods are fixed, changing these quantities has the effect of rescaling the inferred R_t about 1, but does not impact on the other parameters significantly. Lower panel: the deterministic SEINRD R code captures the first, second and third wave of the outbreak in terms of cases and deaths attributed to COVID-19. (Online version in colour.)

of research outputs. We felt that the media storm surrounding some of the scientific advice during the pandemic demonstrated that such a tool would be valuable in improving trust in science used for public policy. In addition to the indirect and intangible costs arising from diminished public faith in science, there is also potential for wider efficiency benefits to accrue from developing and using software to support FAIR data management. An analysis of the qualitative and quantitative costs of not having FAIR data has estimated a negative impact of €10.2bn on the European economy [81].

Accordingly, a requirements analysis was run to determine what capabilities were needed from such a tool—specifically, what functionality a data pipeline would need to provide to be useful to not just to the epidemiological and other modelling communities, but also the broader lay and policy audience [33]. After a review of software available for data management and reproducible research, we concluded that no suitable tools existed in the public domain that could be easily adapted to these uses. We therefore developed the new open source suite of tools for FAIR management of data and models described here to improve the openness of science for policy and better support the traceability of evidence, with as low a barrier of entry as we could devise. Although a small amount of friction remains, our intention has been to ensure that there are benefits for all potential users that can be realized in only a few minutes: for instance, for modellers, the ability to automatically trace the provenance of the output of an existing model with only a single short configuration file to describe any input and output data, and minimal changes to the code; for a data manager or user to look up a file in a data registry and see what (if any) metadata is held on it, and what other versions of that data product also exist; or for third parties, the ability to look up a model output in the same data registry and see its provenance immediately, linked directly back to the code and data used to create it.

Critically, these tools are as non-prescriptive as possible in terms of how, in what environments, and with what software users carry out their analyses and modelling, while nonetheless tracking in fine detail exactly what versions of what data are ingested, what commits of what software are run and by whom, and what research outputs are generated. As well as automatically generating detailed provenance information, the pipeline also annotates the runs with other detailed metadata, and relies on existing standards to maximize the FAIRness of the data produced and to enable interoperability with other resources. Finally, where possible, it reduces the burden on users of correctly annotating data provided from external data providers (or even their own files) by retrieving metadata (and even the data itself) directly from the sources. Where this is not possible, it provides a simple text-based format to manually upload the necessary information in a straightforward manner.

Isolating metadata completely from the data being analysed and used means that this resource (in the form of data registries that can themselves be run directly on user laptops or created in the cloud and shared across research groups if desired) can be made publicly available even when the underlying data are highly sensitive, as usually the metadata and data have different licenses. When allowed by the metadata license, we can include it in the registry since the data are only identified through the checksums of the files (to ensure traceability) and any other metadata that the user chooses to upload.

By using standard formats and vocabularies, we are maximizing the FAIRness of the metadata stored in the registry, enabling interoperability with other resources. This also allows us to export the metadata in recognized and proven formats with existing tools to manipulate them, and the process of aggregating data should be simplified. However, there are still many challenges when including external resources and aggregating the metadata. Where the data are openly available, some sources provide data (e.g. CSV files) without any description of what the data means. Other sources provide data dictionaries, and this improves our interpretative ability when mapping the data into our registry, but, nevertheless, the mapping process is manual, error prone and time consuming. By promoting a more standardized approach, we hope that data providers in epidemiological modelling and other domains may follow our lead in adopting the standards for data description that we support, which are widely used in other application areas.

8. Conclusions

(a) Trust, but verify

By enabling the public release of the provenance of scientific policy advice, we believe that this data pipeline will allow users to be open about their work in a way that can only increase trust in well-founded scientific conclusions. It will increase trust by allowing verification to take place

more easily, and it will allow users to more easily identify potential problems in the logical constructs that have led to their conclusions through the integration of the issue tracker into the provenance system. With climate change an ongoing crisis, subject to sectional political and wider societal argument, and with critical inputs to any solution needing to come from the scientific community, the COVID-19 pandemic will not be the last time that we will be challenged on our openness and trustworthiness.

A situation where a domain expert delivers policy-relevant model-derived evidence, either without all of the choices made in generating this evidence being made explicit or without providing supporting evidence for these choices, is clearly problematic in terms of public trust. In practice, in the short term, there may be insufficient time to deliver either comprehensively. Alternatively, where decisions have to be made rapidly, potentially based on uncertain and rapidly changing information, a key functionality is to ensure that choices are documented, to facilitate ongoing assessment of their validity, by the modellers themselves as part of their own scientific processes, but also by a wider population of scientific peers, and to a more limited but still valuable extent, by science-policy brokers, policymakers and the wider general public. Although the latter will not necessarily have the time or technical background to assess the technical validity of modelling and analysis assumptions, trust can nevertheless indirectly be promoted by facilitating technical scientific challenge by those who are most equipped to do so.

It is also desirable that, as part of the provision of expert advice, domain experts are facilitated in curating a record of the choices they have made, and of their evolving understanding of these. In particular, ongoing assessment of the validity and relevance of model releases and key data resources should be facilitated, whether by the modeller or by informed stakeholders. In particular, where advice changes in the light of new information, support for resulting shifts in policy will be enhanced by an ability to demonstrate the link between new assumptions and new conclusions. Other, more immediately pragmatic requirements include the ability to identify analyses and outputs that depend on outdated code or data; an ability to validate or invalidate a past analysis conditional on the current status of its underlying assumptions and data; and an ability to retrospectively reconstruct the validity of an analysis at a previous point in time, given the status of the underlying assumptions at that time. We anticipate that these key functions will be particularly important to those specialists brokering evidence across the science-policy interface. Where the domain expert makes metadata detailing the provenance of a dataset or codebase available, the interpretation of these in terms of applicability and validity can be assessed and challenged by their technically proficient peers. So long as the domain expert is updating the data registry to reflect changes in their own perception of the validity of assumptions and data resources, any other user can use the data pipeline to meet the operational needs described earlier or simply to explore the links between outputs and assumptions over time. Thus, in general, the use of the data pipeline facilitates good curation of metadata by the modeller, supports informed evaluation of work by scientific peers and democratizes access by the wider community to information about model assumptions. In so doing, we believe that a tool such as the data pipeline can help maintain public confidence in scientists and scientific work at the high level which best supports society and its needs.

(b) Future work

(i) Documentation and validation

The plan to operationalize the pipeline is to integrate a suite of realistic policy-oriented models into the data pipeline. The use cases previously described in §2a, detailing a wide range of activities likely to be carried out by identified users (including mathematical modellers, science-policy brokers, policymakers and the wider public), will each be implemented for the integrated mathematical models, as part of a process of analysing user–software interactions and developing documented procedures. We would hope to involve science-policy brokers in this process; their

involvement will be invaluable, in that they are a key target user group and can also plausibly serve as proxies for policymakers and the general public in the process. In particular, use cases 9-13 are different inspections of data and results that they (and other individuals like members of the public) might wish to make to understand the origins of conclusions that researchers present. Currently, the data registry's web interface to address these use cases (e.g. electronic supplementary material, figure 2) is limited, but further work is underway to improve this. Tools for provenance visualization are also limited, and we believe further work is needed in this area to reduce the complexity of the diagrams produced, and increase their ease of use for exploration of data and results. If gaps become evident in the portfolio of use cases, these will be documented and carried forward for further attention. In the longer term, we intend to pilot uptake in groups delivering model-based evidence to policy; it is likely that initial implementation and evaluation would best be carried out as part of an emergency simulation exercise, where the utility, costs and robustness of the data pipeline could be assessed within the context of the wider demands made of the scientists by policymakers.

(ii) Metadata automation

At the moment, several aspects of pipeline use are manual where they could be automated. These include (i) the attribution of authors to software, which could be taken from `CITATION.cff` or `.zenodo.json` files, or other metadata in the git repository; (ii) the integration of metadata (including authorship) into the pipeline from DOIs associated with data as it is ingested; (iii) the automatic creation of Issues and CodeRepoReleases in the pipeline from issues raised and releases created in GitHub; (iv) the automatic generation of persistent identifiers (e.g. DOIs) when ExternalObjects are created; and (v) improving the ease of syncing metadata between local and remote registries. Integrating this functionality into the pipeline is desirable, since it will further increase ease of use, reducing the barrier of entry for new users.

(iii) Interoperability

Further work is also necessary on interoperability, to increase the FAIRness of the data managed by the pipeline, in particular to make it easier to catalogue, search, access and reuse metadata. The pipeline can already export metadata in PROV-O and JSON-LD formats for provenance and linked data, with some descriptions of the data products using the DCAT vocabulary. These representations can be extended to include more details on the different research objects as required. In addition, we also need to be able to export the whole registry in a DCAT-compliant way for interoperability with other data catalogues, as well as exporting whole research objects with their provenance and other metadata in RO-Crate format. This should also allow us to import data with associated metadata directly from other platforms, either directly or via specifically created mappings to recognized standards. Finally, we need to integrate the ability to use different storage engines as backing stores for the pipeline.

Data accessibility. All code is available through the GitHub organization FAIRDataPipeline—https://github.com/FAIRDataPipeline—and other materials are available through links within the paper. Code directly related to the data pipeline also has DOIs from Zenodo (see table 1). All code will be published with permanent DOIs on Zenodo before publication. The data are provided in the electronic supplementary material [82].

Authors' contributions. S.N.M.: data curation, methodology, software, supervision, validation, visualization, writing—review and editing; A.L.: data curation, methodology, software; J.H.: data curation, methodology, software; N.C.: data curation, software, visualization, writing—review and editing; B.B.: software, validation; R.F.: software, validation; D.R.: software; K.Z.: methodology, software; A.W.: methodology, software, visualization, writing—review and editing; B.V.: software; M.B.: software; B.A.: software; P.B.: software; R.T.B.: methodology, software; L.A.B.: funding acquisition, methodology; A.B.: conceptualization, funding acquisition, project administration, supervision, writing—original draft, writing—review and editing; S.P.B.: software; R.D.: conceptualization, data curation, funding acquisition, writing—original draft, writing—review and editing; J.E.: methodology, software; A.G.: conceptualization, data curation, funding acquisition, methodology, writing—original draft, writing—review and editing; C.H.: software;

I.H.: software, supervision; C.D.H.: methodology, software; M.K.: software, validation; V.M.: software; G.M.: conceptualization, funding acquisition, supervision, validation, writing—original draft, writing—review and editing; L.M.: conceptualization, funding acquisition, writing—original draft, writing—review and editing; I.M.: conceptualization, data curation, funding acquisition, methodology, project administration, supervision, writing—original draft, writing—review and editing; C.M.: data curation; D.J.M.: conceptualization, funding acquisition; S.M.: software; C.M.P.: validation; T.P.: software; A.R.: conceptualization; E.T.: software; R.T.: conceptualization, funding acquisition, project administration, supervision, writing—original draft, writing—review and editing; J.W.: data curation, writing—review and editing; R.R.: conceptualization, funding acquisition, methodology, project administration, software, supervision, writing—original draft, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. The work was funded by the Science and Technology Facilities Council under grant no. ST/V006126/1, Biotechnology and Biological Sciences Research Council (grant nos. BB/M003949/1, BB/R012679/1 and BB/S001034/1), Engineering and Physical Sciences Research Council (grant nos. EP/T004878/1 and EP/V054236/1), Medical Research Council (grant nos. MC_UU_00022/2 and MR/R00241X), Natural Environment Research Council (grant nos. NE/T004193/1 and NE/T010355/1), the Scottish Government Rural and Environment Science and Analytical Services Division (grants 'Centre of Expertise in Animal Disease Outbreaks' and 'Strategic Research Programme'), Scottish Government Chief Scientist Office (grant SPHSU17), the UK Atomic Energy Authority, supported by BEIS, the French National Research Agency (ANR) (IDEXLYON project, grant no. ANR-16-IDEX-0005) and Boehringer Ingelheim Animal Health France (The Veterinary Public Health (VPH) hub).

Acknowledgements. This work was initially undertaken as a contribution to the Rapid Assistance in Modelling the Pandemic (RAMP) initiative, coordinated by the Royal Society. We are very grateful to all of the volunteers in the Scottish COVID-19 Response Consortium formed as part of that initiative – https://www.gla.ac.uk/scrc – who contributed to this project, and in particular to the team at Man Group for their assistance in designing and helping to build the initial version of this pipeline through the Royal Society's RAMP initiative, and the team at Invenia Labs – Eric Perim Martins, Bella Wu, Sean Lovett and Alex Robson – for their invaluable assistance with the Julia implementation.

References

1. Centre for Mathematical Modelling of Infectious Diseases. 2021 COVID-UK. London School of Hygiene & Tropical Medicine. Original date: 2020-05-04T16:42:32Z. See https://github.com/cmmid/covid-uk.
2. OTHERINFOMRC Centre for Global Infectious Disease Analysis. CovidSim. MRC Centre for Global Infectious Disease Analysis; 2021. Original date: 2020-05-04T16:42:32Z. See https://github.com/mrc-ide/covid-sim.
3. OTHERINFOAdam Kucharski. 2020-cov-tracing. London School of Hygiene & Tropical Medicine; 2021. Original date: 2020-05-04T16:42:32Z. See https://github.com/adamkucharski/2020-ncov.
4. OTHERINFOAdam Kucharski. 2020-ncov. London School of Hygiene & Tropical Medicine; 2021. Original date: 2020-05-04T16:42:32Z. See https://github.com/adamkucharski/2020-ncov.
5. Kucharski AJ *et al.* 2020 Early dynamics of transmission and control of COVID-19: a mathematical modelling study. *Lancet Infect. Dis.* **20**, 553–558. (doi:10.1016/S1473-3099(20)30144-4)
6. Giordano G, Blanchini F, Bruno R, Colaneri P, Di Filippo A, Di Matteo A, Colaneri M. 2020 Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy. *Nat. Med.* **26**, 855–860. (doi:10.1038/s41591-020-0883-7)
7. Kucharski AJ *et al.* 2020 Effectiveness of isolation, testing, contact tracing, and physical distancing on reducing transmission of SARS-CoV-2 in different settings: a mathematical modelling study. *Lancet Infect. Dis.* **20**, 1151–1160. (doi:10.1016/S1473-3099(20)30457-6)
8. Davies NG, Barnard RC, Jarvis CI, Russell TW, Semple MG, Jit M, Edmunds WJ. 2021 Association of tiered restrictions and a second lockdown with COVID-19 deaths

and hospital admissions in England: a modelling study. *Lancet Infect. Dis.* **21**, 482–492. (doi:10.1016/S1473-3099(20)30984-1)

9. Keeling MJ. 2005 Models of foot-and-mouth disease. *Proc. R. Soc. B* **272**, 1195–1202. (doi:10.1098/rspb.2004.3046)
10. Matthews L, Haydon DT, Shaw DJ, Chase-Topping ME, Keeling MJ, Woolhouse MEJ. 2003 Neighbourhood control policies and the spread of infectious diseases. *Proc. R. Soc. B* **270**, 1659–1666. (doi:10.1098/rspb.2003.2429)
11. OSR Statement regarding transparency of data related to COVID-19. 2020. See https://osr.statisticsauthority.gov.uk/news/osr-statement-regarding-transparency-of-data-related-to-covid-19/.
12. Wilkinson MD *et al.* 2016 The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 160018. (doi:10.1038/sdata.2016.18)
13. Lamprecht AL *et al.* 2020 Towards FAIR principles for research software. *Data Sci.* **3**, 37–59. (doi:10.3233/DS-190026)
14. Goble C, Cohen-Boulakia S, Soiland-Reyes S, Garijo D, Gil Y, Crusoe MR, Peters K, Schober D. 2020 FAIR computational workflows. *Data Intell.* **2**, 108–121. (doi:10.1162/dint_a_00033)
15. FAIRplus. FAIR Cookbook;. See https://fairplus.github.io/the-fair-cookbook/content/home.html.
16. Jacobsen A, Kaliyaperumal R, Mons B, Schultes E, Roos M, Thompson M. 2020 A generic workflow for the data FAIRification process. *Data Intell.* **2**, 56–65. (doi:10.1162/dint_a_00028)
17. Davidson LA, Douglas K. 1998 Digital object identifiers: promise and problems for scholarly publishing. *J. Electron. Publ.* **4**. (doi:10.3998/3336451.0004.203)
18. ORCID. See https://orcid.org.
19. JSON-LD - JSON for Linking Data. See https://json-ld.org/.
20. Cox SJD, Gonzalez-Beltran AN, Magagna B, Marinescu MC. 2021 Ten simple rules for making a vocabulary FAIR. *PLoS Comput. Biol.* **17**, e1009041. (doi:10.1371/journal.pcbi.1009041)
21. Scottish COVID-19 Response Consortium. 2020 Scottish COVID-19 Response Consortium. See https://www.gla.ac.uk/scrc.
22. Royal Society. 2020 Rapid assistance in modelling the pandemic: RAMP. See https://royalsociety.org/topics-policy/Health.
23. Groth P, Moreau L. 2013 PROV-Overview. See https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/.
24. Albertoni R, Browning D, Cox S, Gonzalez-Beltran A, Perego A, Winstanely P. W3C-DCAT. See https://www.w3.org/TR/2020/REC-vocab-dcat-2-20200204/.
25. European Organization For Nuclear Research, OpenAIRE. Zenodo. CERN; 2013. See https://www.zenodo.org/.
26. figshare. figshare; 2021. See https://figshare.com.
27. Wolstencroft K *et al.* 2016 FAIRDOMHub: a repository and collaboration environment for sharing systems biology research. *Nucleic Acids Res.* **45**(D1), D404–D407. (doi:10.1093/nar/gkw1032)
28. Wolstencroft K *et al.* 2015 SEEK: a systems biology data and model management platform. *BMC Syst. Biol.* **9**, 33. (doi:10.1186/s12918-015-0174-y)
29. Splitgraph. See https://www.splitgraph.com.
30. Dolt is Git for Data! DoltHub; 2021. Original date: 2019-07-24T17:46:25Z. See https://github.com/dolthub/dolt.
31. DoltHub Home. See https://www.dolthub.com/.
32. data.world | The Cloud-Native Data Catalog. See https://data.world/.
33. Scottish COVID-19 Response Consortium. 2021 Data Pipeline Use Cases. See https://www.fairdatapipeline.org/docs/use_cases/.
34. Freire J, Koop D, Santos E, Silva C. 2008 Provenance for computational tasks: a survey. *Comput. Sci. Eng.* **10**, 11–21. (doi:10.1109/MCSE.2008.79)
35. Afgan E *et al.* 2018 The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res.* **46**, 537–544. (doi:10.1093/nar/gky379)
36. Apache Airflow. See https://airflow.apache.org.
37. Kepler. See https://kepler-project.org/.
38. Kaggle: Your machine learning and data science community. See https://www.kaggle.com/.
39. Jupyter. See https://jupyter.org/index.html.

40. Information Commissioner's Office. Guide to the UK General Data Protection Regulation (UK GDPR); 2021. See https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/.
41. Quilt. See https://github.com/quiltdata/quilt.
42. Docker. See https://www.docker.com.
43. QuiltData. See https://quiltdata.com.
44. FloydHub Blog. See https://blog.floydhub.com/.
45. Covid Model-Runner. COVID-19 Modeling; 2020. Original date: 2020-05-04T16:42:32Z. See https://github.com/covid-modeling/model-runner.
46. Williamson EJ *et al.* 2020 Factors associated with COVID-19-related death using OpenSAFELY. *Nature* **584**, 430–436. (doi:10.1038/s41586-020-2521-4)
47. OpenSAFELY: Home. See https://www.opensafely.org/.
48. Davidson SB, Freire J. 2008 Provenance and scientific workflows: challenges and opportunities. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1345–1350. Vancouver, Canada: Association for Computing Machinery, New York, NY, United States of America.
49. Pasquier T *et al.* 2017 Practical Whole-System Provenance Capture. In *Proceedings of the Symposium on Cloud Computing (SoCC'17), Santa Clara, CA, Sept*, pp. 405–418. Association for Computing Machinery. (doi:10.1145/3127479.3129249)
50. Pohly D, Mclaughlin S, McDaniel P, Butler K. 2012 Hi-Fi: Collecting high-fidelity whole-system provenance, pp. 259–268.
51. Lerner B, Boose E, Perez L. 2018 Using introspection to collect provenance in R. *Informatics* **5**, 1–18. (doi:10.3390/informatics5010012)
52. Jones MB, Slaughter P, Jones C. recordr. See http://github.com/NCEAS/recordr.
53. Jackson M *et al.* 2018 recipy. See https://github.com/recipy/recipy.
54. Core Provenance Library. See https://github.com/ProvTools/prov-cpl.
55. Git. See https://git-scm.com/.
56. git-annex. See https://git-annex.branchable.com/.
57. Qri. See https://qri.io/.
58. Pachyderm. See https://www.pachyderm.com/.
59. Kubernetes. 2021. See https://kubernetes.io.
60. DVC. See https://dvc.org/.
61. DAGsHub. See https://dagshub.com/.
62. MLflow. See https://mlflow.org.
63. Halchenko YO *et al.* 2021 DataLad: distributed system for joint management of code, data, and their relationship. *J. Open Sourc. Softw.* **6**, 3262. (doi:10.21105/joss.03262)
64. Sefton P *et al.* 2021 RO-Crate Metadata Specification 1.1.1. NA. Zenodo. See https://zenodo.org/record/4541002.
65. Scottish COVID-19 Response Consortium. 2020 software-checklist. See https://github.com/ScottishCovidResponse/modelling-software-checklist.
66. Scottish COVID-19 Response Consortium. 2021 FAIRDataPipeline GitHub Organisation. See https://github.com/FAIRDataPipeline.
67. Blackwell R *et al.* 2021 The FAIR Data Registry. Zenodo. See https://doi.org/10.5281/zenodo.5562749.
68. Zarebski K, Reeve R, Reddyhoff D, Cummings N. 2021 The FAIR Data Pipeline command line tool. Zenodo. See https://doi.org/10.5281/zenodo.5552779.
69. Mitchell S, Field R. 2021 rDataPipeline - FAIR Data Pipeline in R. Zenodo. See https://doi.org/10.5281/zenodo.5338588.
70. Mitchell S, Burke M, Reeve R. 2021 DataPipeline.jl - FAIR data pipeline in Julia. Zenodo. See https://doi.org/10.5281/zenodo.5270282.
71. Boskamp B. 2021 javaDataPipeline - FAIR Data Pipeline in Java. Zenodo. See https://doi.org/10.5281/zenodo.5547492.
72. Field R, Reddyhoff D. 2021 pyDataPipeline - FAIR Data Pipeline in Python. Zenodo. See https://doi.org/10.5281/zenodo.5548002.
73. Field R, Zarebski K. 2022 C++ Implementation of the API for the FAIR Data Pipeline. Zenodo. See https://zenodo.org/record/5877992.

74. Bjørnstad ON, Shea K, Krzywinski M, Altman N. 2020 The SEIRS model for infectious disease dynamics. *Nat. Methods* **17**, 557–558. (doi:10.1038/s41592-020-0856-2)
75. Mitchell S. 2021 rSimpleModel. See https://github.com/FAIRDataPipeline/rSimpleModel.
76. Mitchell S. 2021 SEIRS Model Example. See https://www.fairdatapipeline.org/rSimpleModel/articles/SEIRS.html.
77. Boskamp B. 2021 javaSimpleModel. See https://github.com/FAIRDataPipeline/javaSimpleModel.
78. Mitchell S. 2021 SEINRD Model Example. See https://www.fairdatapipeline.org/rSimpleModel/articles/SEINRD.html.
79. Offices of the Nuffield Professor of Medicine. COVID-19 Infection Survey; 2021. See https://www.ndm.ox.ac.uk/covid-19/covid-19-infection-survey.
80. Pooley CM, Doeschl-Wilson AB, Marion G. 2022 Estimation of age-stratified contact rates during the COVID-19 pandemic using a novel inference algorithm. *Phil. Trans. R. Soc. A*
81. European Commission Directorate General for Research and Innovation, PwC EU Services. Cost-benefit analysis for FAIR research data: cost of not having FAIR research data. LU: Publications Office; 2018. See https://data.europa.eu/doi/10.2777/02999.
82. Mitchell SN *et al.* 2022 FAIR data pipeline: provenance-driven data management for traceable scientific workflows. Figshare. (doi:10.6084/m9.figshare.c.6070465)