Openness and computational reproducibility in plant pathology: where we stand and a way forward

Adam H. Sparks1,2,✉, Emerson M. Del Ponte3, Kaique S. Alves3, Zachary S. L. Foster4, and Niklaus J. Grünwald4

1 Farming Systems Innovation, Department of Primary Industries and Regional Development, Perth WA 6000, Australia  
2 University of Southern Queensland, Centre for Crop Health, Toowoomba Qld 4350, Australia  
3 Departmento de Fitopatologia, Universidade Federal de Viçosa, Brazil  
4 Horticultural Crops Disease and Pest Management Research Unit, USDA Agricultural Research Service, Corvallis OR 97330, USA

✉ Corresponding author: [Adam H. Sparks <[Adam.Sparks@dpird.wa.gov.au](mailto:Adam.Sparks@dpird.wa.gov.au)>](mailto:Adam.Sparks@dpird.wa.gov.au)

**Keywords: Data Science, Computational Biology, Techniques**

# Abstract

Open research practices have been highlighted extensively during the last ten years in many fields of scientific study as essential standards needed to promote transparency and reproducibility of scientific results. Scientific claims can only be evaluated based on how protocols, materials, equipment and methods were described; data were collected and prepared; and analyses were conducted. Openly sharing protocols, data and computational code is central for current scholarly dissemination and communication, but in many fields, including plant pathology, adoption of these practices has been slow. We randomly selected 450 articles published from 2012 to 2021 across 21 journals representative of the plant pathology discipline and assigned them scores reflecting their openness and computational reproducibility. We found that most of the articles were not following protocols for open science and were failing to share data or code in a reproducible way. We propose that use of open-source tools facilitates computationally reproducible work and analyses benefitting not just readers, but the authors as well. Finally, we provide ideas and suggest tools to promote open, reproducible computational research practices among plant pathologists.

# Introduction

Modern plant pathological research has many facets given the array of disciplines and sub-disciplines currently involved. Collectively, they contribute to increasing our basic and applied knowledge of several aspects of pathogen biology and disease development to ultimately improve plant disease management. Scientific research in the field varies from the purely observational or descriptive in nature to inferential (based on experimental or simulation-derived data sets). Whatever the case, the verifiability of research findings depends on how much of the research materials, processes and outcomes are made available beyond what is reported in the scientific article and the ability of others to make use of the methods and results. Examples of such resources include biological materials (host and pathogen genotypes), nucleic/protein sequences, experimental and simulated raw data annotations, drawings and photographs and statistical analysis code.

Open science and reproducibility is becoming more mainstream with many funding agencies expecting data to be available upon the conclusion of the research project (e.g., Government of Canada 2016; Noorden 2017; Australian Research Council 2018; European Commission 2022; Gates Foundation 2022). Journals in the field are promoting the sharing of data (Del Ponte 2020), and more scientists are becoming interested in sharing their raw data and even lab notebook contents (Wald 2010) or drawing attention to the lack of code and data that make published work less useful (Barton et al. 2022).

Reproducibility is one component under the umbrella of open science. By proactively practicing open science, scientists increase the chance that their works become more reproducible due to the availability of data and code. That is, open science enables reproducibility and replicability.

To ensure clear communication on this topic, we first must define terms like reproducibility. Many of the terms used in this area have varying definitions that may or may not agree with each other. For instance, reproducible research was recently highlighted by many authors (Tiwari et al. 2021; Eckert et al. 2020; Dienlin et al. 2020; Preeyanon et al. 2018; Wallach et al. 2018; Baker 2016; Iqbal et al. 2016; Editors at Nature 2016; Patil et al. 2016; Weissgerber et al. 2016; Brunsdon 2015; Sweedler 2015; Fitzjohn et al. 2014; Ioannidis 2014; Fidler and Gordon 2013; Stodden et al. 2013) as an important issue.

However, in the biological sciences it is not always possible to use identical test material or environmental conditions, e.g., field trials that span years and locations or complex glasshouse experiments. In other cases there is insufficient time and resources to reproduce the entire study. Therefore, we will follow Peng’s (2009) definition that provides clear guidelines for a minimum standard of “reproducible research”:

“The replication of scientific findings using independent investigators, methods, data, equipment, and protocols has long been, and will continue to be, the standard by which scientific claims are evaluated. However, in many fields of study there are examples of scientific investigations that cannot be fully replicated because of a lack of time or resources. In such a situation, there is a need for a minimum standard that can fill the void between full replication and nothing. One candidate for this minimum standard is ‘reproducible research’, which requires that data sets and computer code be made available to others for verifying published results and conducting alternative analyses”. Peng (2009). *Reproducible research and Biostatistics*. Biostatistics, 10 (3): 405-408.

Therefore, our definition of computational reproducibility will be that the computer code and data are made freely available to others for verification and conducting alternate analyses or for use in instructional purposes. In addition, the software used should be easily obtained and preferably open-source to avoid licensing or other issues related to accessibility for end-users related to costs or non-standard file formats.

Plant pathologists often provide information on protocols and chemicals allowing for reproducibility. However, biological specimens such as strains, cultures or cultivars are often not available after publication. These cases do constitute a lack of reproducibility but will not be covered here.

Plant pathologists often provide information on protocols and chemicals allowing for reproducibility. However, frequently biological specimens such as strains, cultures or cultivars are not available after publication. These cases do constitute a lack of reproducibility but will not be covered here.

# Materials and Methods

## Article Selection and Evaluation Events

To understand where we stand as a discipline regarding open science and reproducible research, we surveyed a broad selection of articles representing a wide swath of publications to evaluate our collective status. We hand-picked 21 journals which represented research publications in the field of plant pathology (Table 1) that encompassed a range of subject matter foci, including applied and fundamental work, country of origin and ranking metrics, e.g., quartile range or citation index. The aim was to gather as complete an overview of the status of computational reproducibility in plant pathology journals as possible and avoid bias in the findings by skewing towards high-impact journals that may have a greater influence. From those 21 journals we randomly selected 450 articles published from 2012 to 2021. Using R (R Core Team 2022), two lists were created. The first was a list of the 21 journals and the second was a list of the evaluators that were evaluating the articles for reproducibility. Initially there were four evaluators, later a fifth was added and the list was recreated. There were three evaluation events that took place to increase the yearly coverage of the evaluations. During each scoring event, each evaluator was randomly assigned 50 unique articles to evaluate.

A list of randomly generated numbers representing page numbers from one to 150, sampled with replacement, was assigned to a randomized list of 21 journals for each sampling event. This was done to ensure that the randomly generated number had a corresponding number in the journal.

A Google Sheets file was created to hold the article details and metadata for the paper’s authors to refer in finding their assigned articles. The five-year impact factor for 2022 for each journal was retrieved from InCites Journal Citation Reports, Clarivate and entered in the spreadsheet with the article details for the respective journals. The authors entered article scores and comments about the article in a separate spreadsheet in the Google Sheets file to keep the original details intact and avoid overwriting or making possible errors that would lead to the loss of data. The entire workbook was downloaded and saved as an Open Document Spreadsheet (ODS) file for analysis after all the values were recorded.

Articles were then manually selected by visiting the journals’ websites and selecting the articles within which the randomly assigned page numbers fell, *i.e.*, if the page number was 32, the article that started on page 28 and ended on page 35 was selected as it contained page 32. In cases where an article was not suitable, e.g., a review or otherwise not related to plant pathology or the randomly assigned numbers for that journal and year fell within the same article, the next article in that journal was selected until a suitable article was found.

During three scoring events, two to five of this paper’s authors were each assigned to rate a randomized list of journal articles using scoring criteria devised for the purposes of this research as their time allowed for their participation. In the first event, three authors (AS, EDP, and ZF) evaluated a total of 200 unique articles, in the second event two authors evaluated a total of 100 unique articles (EDP, KA) and in the third round, five authors (AS, EDP, KA, ZF, NG) evaluated a total of 150 unique articles for a total of 450 articles. Each article was evaluated by only one evaluator save five articles randomly selected for inter-rater repeatability analysis from the third and final evaluation event.

## Scoring Criteria

Each article was rated on a 0–3 scale for its data and code availability. ‘Code availability’ rated how easily and openly available the computational methods used in the article were. Scores were assigned as ‘0’ was ‘Not available or not mentioned in the publication’; ‘1’ was ‘Available upon request to the author; ’2’ was ‘Online, but inconvenient or non-permanent (e.g., login needed, paywall, FTP server, personal lab website that may disappear, or may have already disappeared)’; and ‘3’ was ‘Freely available online to anonymous users for foreseeable future (e.g., archived using Zenodo, dataverse or university library or some other proper archiving system)’; ‘NA’ indicates that no code was created to conduct the work that was detectable. Second, the ‘Data availability’ evaluated how freely and openly available the data presented in the article were. This was evaluated as ‘0’ was ‘Not available or not mentioned in the publication’; ‘1’ was ‘Available upon request to the author; ’2’ was ‘Online, but inconvenient or non-permanent (e.g., login needed, paywall, FTP server, personal lab website that may disappear, or may have already disappeared)’; and ‘3’ was ‘Freely available online to anonymous users for foreseeable future (e.g., archived using Zenodo, dataverse or university library or some other proper archiving system)’; ‘NA’ indicates that no data were generated, e.g., a methods paper.

For the purposes of this research paper, the word “code” in “code availability” was defined as including any software that were used in the data import, cleaning, analysis, genome assembly or manuscript preparation or in the case of modelling papers, the model’s software code itself. The word “data” in “data availability” was defined as including any data created or collected and presented in the research manuscript.

In cases where partial computational materials or data were made available, the score was assigned the lowest score reflecting the completeness of the availability, all or nothing. That is, in cases where some data such as sequences were deposited in a database, e.g., GenBank, but other data that was collected and used were not made available, the score assigned was ‘0’.

Where possible, the software used in conducting the research behind the publication was recorded in the notes when it was cited or otherwise specified in the article text.

## Data Cleaning

The criteria scores were checked for inconsistencies using the evaluators’ comments in the spreadsheet and based on the software used. In some cases these comments were used to adjust the criteria score to align with the scoring definitions. This was not possible in all cases but only in cases where the evaluator had left comments that could be checked and acted upon. For example, a reviewer noted that all data were deposited in GenBank but the data score was ‘0’, this was changed to ‘3’ to align with the definition. Or in other cases, some software packages used were Excel add-ins, but Excel was not recorded as being used; Excel was added to the list of software used to rectify this.

A custom function, import\_notes(), was written to import the data and format the columns properly in R (R Core Team 2022). Values for software packages were checked for spelling consistency and corrections were made manually where necessary. When working with these data in R, all strings of software character values were converted to fully upper-case to standardize the capitalization and alleviate any issues with capitalization used between evaluators.

## Statistical Analysis

Inter-rater differences were evaluated using percent agreement and Fliess’ Kappa (Del Ponte et al. 2019).

Six models were applied, with half predicting the response of code availability and the other half predicting data availability. Models 1 and 2 were used to predict responses with publishing journal, using a factor, the journals’ abbreviation, as the predictor (formula: response ~ abbreviation). Models 3 and 4 were used to predict the two responses with a continuous value, year, as the predictor (formula: response ~ year) and the models included abbreviation and assignee as random effects (formula: list(~1 | abbreviation, ~1 | assignee)) to detect changes that occurred over the years in which the manuscripts were published that were included. Models 5 and 6 were used to predict the responses using a continuous value, IF\_5year, as the predictor, (formula: response ~ IF\_5year) and included assignee as random effect (formula: ~1 | assignee) without journal as a random effect to mitigate correlation between impact factor and the journal of publication. All models included the assigned article evaluator as a random effect (formula: ~1 | assignee) to address inter-rater effects on scoring.

We fit Bayesian logistic mixed models (estimated using MCMC sampling with 4 chains of 10000 iterations and a warm-up of 5000 and thinning of 1) using cumulative family function with a logit link for ordinal data using the contributed package ‘brms’ (version 2.18.0) (Bürkner 2017, 2018, 2021). The cumulative logit family is often used for ordinal data as it uses cumulative probabilities up to a threshold, making all of the ordinal categories binary at that threshold. Priors were selected to be weakly informative and deemed suitable through using pp\_check() to examine the models’ predictions based on priors only (using the parameter sample\_prior = only in brm()). Priors over all parameters were set as: normal (mean = 0.00, SD = 1.0) distributions for both fixed parameters and random effects. The best fitting models were selected by using Widely Applicable Information Criterion (WAIC) and the expected log point-wise predictive density via loo\_compare().

The adapt\_delta() and max\_treedepth() values were adjusted as necessary on a case-by-case basis for each of the models to ensure a good model fit to the data and that the chains mixed well.

The fitness of all models was evaluated using model summaries and diagnostic plots from ‘brms’ (version 2.18.0) and posterior fits using pp\_check() from the contributed R package ‘bayesplot’ (version 1.9.0) (Gabry et al. 2019).

A test for practical equivalence, equivalence\_test(), was performed for each model using the contributed R package ‘bayestestR’ (version 0.13.0) (Makowski et al. 2019) with the Region of Practical Equivalence (ROPE) set to -0.1 to 0.1 and confidence interval of 0.95.

Using the Sequential Effect eXistence and sIgnificance Testing (SEXIT) framework, the median of the posterior distribution and its 95% CI (Highest Density Interval), along the probability of direction (pd), the probability of significance and the probability of being large are reported. The thresholds beyond which the effect is considered as significant (i.e., non-negligible) and large are 0.05 and 0.3 as suggested by Makowski et al. (2019). Convergence and stability of the Bayesian sampling was assessed using R-hat, which should be below 1.01 (Vehtari et al. 2021) and Effective Sample Size (ESS), which should be greater than 1000 (Bürkner 2017).

A full report of all parameters and model details are available in the Supplementary Materials generated using the contributed package ‘report’ (version 0.4.0) (Makowski et al. 2021). Additionally, all methods are described in greater detail with the code necessary to reproduce the work in Sparks et al. (2022).

All statistical analysis was performed using R version 4.2.1 (2022-06-23) (R Core Team 2022) on an Apple MacBook Pro (13-inch, M1, 2020).

# Results

## Inter-rater Agreement

All authors agreed on the five inter-rater article evaluations for the data availability, ‘0’, not available for an inter-rater score of 100%. However, one of the authors rated one article as ‘NA’ or not having any computational methods used rather than ‘0’ as with the other four authors, giving the inter-rater percent agreement a score of 100% with the ‘NA’ value dropped from the code availability.

Looking at the ratings for all 450 articles and given the differing number of articles each co-author evaluated, there was broad agreement with no discernible patterns showing a strong bias or cause for concern (Figure S1). However, the analyses did account for different raters (assignees) for each article by including this as a random effect.

## Code and Data Sharing Findings

Most articles did not make any computational methods available in any fashion with 4 (0.9%) classified as ‘3’, which was the highest score available (Figure ??A). Four hundred and thirty-five (96.7%) were classed as ‘0’ and ten articles (2%) appeared to not use any computational methods.

Additionally, data that supported the articles were mostly unavailable with 364 (81%) scoring ‘0’ where the original data were not available or mentioned in the article. However, more articles, 48 (11%), scored ‘3’ than scored ‘1’ or ‘2’ combined, 36 (8%) with two articles not producing shareable data (Figure ??D).

## Software Used

There were two hundred and fifty-five unique software applications recorded being used in the articles that were evaluated (Table 2). These included desktop programs, web-based software and databases. From the top ten most frequently cited software, the most frequently cited program was MEGA, where different versions were not distinguished for this work. The next three programs were statistical software, SAS (2nd), SPSS (3rd) and R (4th) with two other statistical programs also frequently cited, GenStat and Statistica (tied 8th). The remainder of the top ten programs we found were mostly related to sequence analysis, *i.e.*, Clustal (5th), BLAST (6th), and BioEdit (9th) or phylogenetic analysis MrBayes (8th), tied with the most general-purpose software cited in the top 10, Excel®. In tenth place was FigTree, used for rendering phylogenetic trees. The Python programming language was cited only twice.

## Statistical Analysis

All models’ diagnostic plots indicated good chain mixing and all were deemed suitable for the data by the checking posterior fits (Figure ??).

When predicting the availability of computational methods (code) and availability, all journals were compared with *Phytopathology*® as the reference with latent scores having a mean of zero.

There were no clear differences that the analysis could detect between any of the journals as compared with *Phytopathology*® for the sharing of computational methods (code); the effects of all parameters were undecided (Figure ??A, Table S1).

When predicting the availability of data availability as compared with *Phytopathology*®, publications in one journal, *Phytopathologia Mediterranea*, can be considered clearly more likely to share data than publications in *Phytopathology*®, but no effects were detectable in any other journal title (Figure ??B; Table S2).

The effect of *Phytopathologia Mediterranea* (Median = 1.68, 95% CI [0.78, 2.57]) had a 99.99% probability of being positive (> 0), 99.98% of being significant (> 0.05), and 99.83% of being large (> 0.30). The estimation successfully converged (Rhat = 1.000) and the indices were reliable (ESS = 19017).

There was no clear effect for year when predicting the availability of computational methods as the analysis results were undecided (Figure ??A; Table S3).

However, the year of publication did not have an effect on the availability of data (Figure ??B, Table S4).

The effect of year (Median = 0.08, 95% CI [-8.06e-03, 0.18]) had a 96.19% probability of being positive (> 0), 75.92% of being significant (> 0.05), and 0.00% of being large (> 0.30). The estimation successfully converged (Rhat = 1.000) and the indices were reliable (ESS = 21692).

There were no detectable effects when predicting the effects of the five-year impact factor on the availability of computational methods (code) or data (Figures ??A, ??B; Tables S5, S6).

# Discussion

Except for a few isolated cases, most papers were not fully computationally reproducible. Very few authors chose to share both data and code. More authors shared data due to journal requirements to share sequence data, but other types of data related to field experiments or other laboratory studies were not likely to be shared. Code sharing was extremely rare, but in cases where it was shared, most were included as a part of the journal article’s publication as extra materials rather than through data repositories. Authors publishing in *Phytopathologia Mediterranea* were more likely than authors publishing in *Phytopathology*® to share data. Several articles published in *Phytopathologia Mediterranea* either made the data available through GenBank or it was noted that all data were available through the article and associated supplementary materials, which may be the reason for this effect.

The reasons for not sharing code or data were not clearly available from the papers themselves and so this work was unable to determine possible reasons for this situation. We recognize that in some cases there may be commercial or intellectual property (IP) reasons for not sharing data or code, but these reasons should be clearly stated. However, in most cases, the data are collected with public funding and the code was developed using similar funding and there are no commercial or IP issues that preclude sharing and in fact the funding agency may have guidelines in place for sharing these materials.

Much of the software used are Free open-source Software (FOSS) packages, which means that the workflow can easily be recreated by anyone with the proper skills in using the software. However, the top three most widely used software packages were not FOSS, limiting the ability of the authors to share workflows with others. MEGA was the most widely cited software in this evaluation reflecting the widespread use of sequence analysis in the field of plant pathology. The second, third and fourth, and eighth most frequently cited software packages were all statistical programs. SAS remains firmly entrenched in the discipline as the choice of software for statistical analysis followed by SPSS but R, the most frequently occurring FOSS, is not far removed from SPSS in fourth place with Statistica and Genstat tied for eighth. It is interesting to note however, that even with the popularity of Python in scientific programming it only appears in the survey twice in 2020 for both articles, not enough times to be in the top 10 most-cited.

## Limitations to Our Approach

We acknowledge that the number of article that are cross-scored is low. However, our aim was not to test inter-rater reproducibility, this is covered at length in the plant pathology literature. Our goal was to evaluate as many articles as we realistically could to get as broad an overview of the discipline as possible. Because of this each article, save the five that were used to compare the inter-rater differences, was evaluated by only one co-author. Therefore, we concluded that the inter-rater agreement was good enough for this paper to illustrate that code and data are not being widely shared with peer-reviewed publications.

Furthermore, the methods for selecting articles meant that only page numbers between 1 and 150 were selected. This approach assumed that there was no effect or bias on reproducibility based on the time of year that an article was published since most journals start with page number one at the beginning of the year and some journals, e.g., *Phytopathology*® have focus issues with invited authors. Despite some publications being from special issues, there were no detectable differences in the reproducibility of the articles as defined and measured by this article, so the assumption appears to be correct and more importantly, may show that even when authors are specifically recruited that they are no more likely to share the supporting materials for the articles that they write.

## Examples and Efforts from Other Disciplines

Plant pathology is not alone in the lack of reproducibility and the closed-nature of data and code.

Seibold et al. (2021) carried out a similar study on the types of methods and software used in PLOS ONE articles and whether the were able to reproduce the data analysis using open source software for papers that dealt with longitudinal data and used mixed effect models or generalized estimating equations for analysis and provided the data. They found that most articles provided tables and visualizations only. However, unlike our analysis, even with the data being supplied they found that only one article used open source software, whereas R prominently featured in the articles we examined along with other free open-source software packages. Not surprisingly they found that replication was mostly difficult and required the results to be reverse engineered or contacting the authors. They were unable to reproduce the results for for three articles and only parts of another two. They needed to contact the articles’ authors for all articles save two. Their main conclusion was that reproducing papers is difficult if no code is provided and puts an undue burden on those interested in reproducing the work.

Other disciplines related to plant pathology have published guides or started communities to promote and support tools and methods that promote good practices that lead to open and reproducible research.In entomology Wittman and Aukema (2020) published “A Guide and Toolbox to Replicability and Open Science in Entomology”, which documented both the benefits and drawbacks of an open and reproducible workflow. Weed scientists started a community to promote openness in the discipline, Open Weed Science, which stated that they valued, “open access knowledge-sharing as a strategy to enhance reproducibility within our discipline”, but the website is no longer available and the Twitter account, @OpenWeedSci, appears to be inactive when viewing the account as of October 2022 (Science and Oliveira 2021). Serra da Cruz and Pires do Nascimento (2019) proposed a framework for capturing provenance of data-centric agronomic experiments using an R-based workflow, ‘RFlow’, which would allow reuse of R scripts using generic workflows and saving data and metadata to web repositories that other researchers and referees can browse using Web interfaces. Kool et al. (2020) evaluated the reproducibility of on-farm experiments in Africa, finding that often the experimental design was difficult or impossible to reproduce from the publication. Fernandez et al. (2022b) provide an R package, ‘cndcR’ (Fernandez 2022), that supports the manuscript: “Dataset characteristics for the determination of critical nitrogen dilution curves: from the past to new guidelines”, (Fernandez et al. 2022a). The package contains source data and R code that Fernandez et al. (2022a) used for their sensitivity analyses when fitting critical N dilution curves in crop species using Bayesian models. The package allows the user to recreate the figures in the manuscript and includes the all of the data files required to produce all results and figures presented in the manuscript. More broadly within agricultural research, articles have been published highlighting the advantages of, need for and how to publish Findable, Accessible, Interoperable and Reusable (FAIR) data in agriculture (Smith et al. 2018; Arnaud et al. 2020; Stacey et al. 2022). However, we were unable to find any quantification of the reproducibility of any of these related disciplines as we have done here.

## Suggestions for Improving Computational Reproducibility

When making science more open and computationally reproducible, methods and software used (e.g. R, Julia or Python packages that were directly used in the analysis or production of the paper, etc.) should be cited properly. Provide a full description of all equipment used, e.g., “a Spectrum Technologies Watchdog 2700 weather station was used to record wind speed, direction, rainfall, temperature and relative humidity at one-hour intervals”. This allows end-users to identify the tools and methods used more accurately. Just as importantly, this acknowledges the contributions of others whose works were instrumental in our research. This also helps to ensure that researchers can reconstruct what they did more easily since good notes and documentation exist and be able to identify if something changes, e.g., a package version, what effect it had on the research.

To help create a good set of data and code that can easily be shared, the use of programming or scripting languages such as R, Julia, SAS or Python enables us to keep very detailed records of what was computationally performed. This is as opposed to using software such as spreadsheet programs like Excel, Google Sheets, Numbers, Calc or others which can be used for simple statistical analyses and visualization or other point-and-click software packages that do not enable us to keep an accurate record of the steps taken to import, format, visualize and analyze data. Text files for saving small sets of data are preferable to proprietary file formats. Data that are saved in binary formats such as PDF files are difficult to reuse because they are not easily machine readable. In many cases, data sets are small enough and curated in spreadsheets, which should be saved as a plain text file, e.g., comma separated (CSV) or tab separated (TSV) files. This also helps to ensure that the data are reusable. Larger data sets may warrant the use of a proper database like lightweight personal databases, e.g., SQLite or DuckDB, larger more robust databases, e.g., MariaDB, PostgreSQL or a specialized database like GenBank which provide users with several benefit. Two important benefits to mention here are (i) avoiding data redundancy, ensuring no records are duplicated and (ii) data consistency, ensuring that all records in a data set are recorded in the same format for every observation. Databases such as GenBank are preferable for molecular data and ensure data integrity and machine readability. While databases may offer many advantages such as speed and data integrity, they are also more complex to set up and administer, and so they may not be the best choice for a small data set.

Ideally, once the data are complete, best practices for keeping our data as we perform our work include treating the raw data as read-only and using file permissions to prevent changes to the raw data files. It should be noted that the use of a database management system also allows for both at the expense of added complexity. Saving files in proprietary formats such as .xls(x) can also lead to issues in the future when opening using newer (or older) software versions. Unexpected changes to values in the data (Ziemann et al. 2016) may also occur when using proprietary formats.

Once these practices are in place, sharing the code and data becomes easier and can be thought of as a layered approach to computational reproducibility.

Proper cataloguing and descriptions of the computational materials and data enable others to easily find and understand what the materials are. Properly constructed metadata allow for interoperability by allowing both humans and machines to understand what the materials are that are included in the resources and make use of them more easily. Metadata also protects the resource and its future availability by tracking the resource’s lineage, describing the resources physical characteristics and behavior so that it can be replicated using future technologies. That is, without metadata, the data are not FAIR and the computational materials lose value by not having good documentation.

If steps are followed to make the data FAIR, then it will be readable by humans and machines alike and this will help support discoveries and support further research. In turn, sharing data can lead to new citations for the work as others discover and use it. To make data the most widely discoverable and usable, ensure that it has a persistent identifier. A digital object identifier (DOI) is the most common (<https://www.doi.org/>) but the Handle.Net Registry (HNR) (<https://handle.net/>) is also an option. There are different options for generating a DOI for data and other materials. FigShare, Zenodo and OSF.io all offer persistent archives along with a service to generate a DOI. The use of a persistent identifier works to ensure that even if the data are moved, they can still be located using that unique identifier. For more on FAIR data, visit Go-Fair <https://www.go-fair.org/fair-principles/>.

Once it has been determined how to best manage the source code and the data sets for analysis, the next step is to consider how to share the data. Providers such as FigShare, Dataverse, OSF.io and Zenodo allow us to deposit our data and provide metadata and generate a DOI for sharing the project once it is finished. Other providers exist that allow for us to not only track changes but also to share the data openly. These include GitHub, Gitlab and Bitbucket. GitHub is arguably the most popular and widely used open-source software development platform currently. However, we would advise against the practice of depositing data on a laboratory website or a site such as GitHub only as these are not an optimal way to preserve and share our work over the long term. Doing either of these leaves the work in an unstable state where future users may be unable to access the work as they are fraught with link-rot and other issues. It is a best practice to always ensure that we have deposited the data with a provider such as GenBank, Zenodo, FigShare or OSF.io and generated a DOI for the materials to help ensure continued accessibility. Many of these providers provide rather easy ways to link the project with a software development repository to help ensure that the data are available in perpetuity. If readers are uncertain, we suggest to also consult with local librarians about possible resources. Most universities and other research-focused workplaces provide a facility for staff to deposit papers and other academic materials, but this may extend to software development or data repositories as well. It is important that data once deposited cannot be modified.

If a general workflow for producing academic research involves defining a research question, obtaining data for testing the hypothesis, analyzing, summarizing and presenting data and results, and writing the manuscript, then the next logical steps are depositing any code used along with the raw data in proper repositories (Figure ??). Here we define three layers of reproducibility which are also related to the evolution of computational methods and reproducible practices and highlight papers that fit these definitions.

The first layer is including tables of raw data or code with the paper as a supplemental file or even within the paper as tables where possible. This is suitable for studies that may have a small data set or simple analysis or for demonstrations purposes as Madden et al. (2015) demonstrate in their discussion regarding the use of values in statistical analyses where they supply an *e-****X****tra* with reproducible examples for readers to refer to. Hill et al. (2019) also shared code and example data to reproduce the results of the paper as supplementary materials, citing all packages used and the versions. The scripts provided used an R package, *checkpoint* (Ooi et al. 2022), to provide a mechanism to help ensure reproducibility by installing the package versions used by the authors that were necessary to reproduce the work rather than defaulting to the latest versions, which may cause failures due to changes in the package code-bases. However, the scripts did not run unhindered as the supplementary materials suggested without changes to the data. But with some modifications to the data, and file name changes, the scripts ran allowing examples of the research to be reproduced using the definition of Peng (2009). One drawback to this approach is that journals are often not equipped to handle code, e.g., script files, that may be developed as a part of the research process and so they are often archived as PDF or Word documents which hinder the ability to easily ingest and start working with them or if they are provided in a native text format, they do not render via the web properly. Though this should be possible given some effort from the publishing journal to share simple text files rather than binary formats along with the proper instructions and handling of these file formats rather than a one-size-fits all approach that we commonly see for supplementary materials. For example, the R scripts for Hill et al. (2019) were provided as supplementary materials but do not actually appear in the browser window when requested, which may confuse readers and makes downloading them more difficult. While this allows the reader to quickly view the extra materials and a DOI is assigned as a part of the article itself, the data or code are not readily findable and accessible through a searchable database. Furthermore, in many cases this does not allow prompt access to the data and running the code because of a journal paywall.

The second layer is providing machine readable text files of the raw data and code in a public code repository such as GitHub, GitLab, but without a DOI or some other long-term repository as provided by an institutional library, Zenodo, FigShare, etc. Fewer authors choose to follow this method, but in one instance that we found, which was not a part of this analysis, Vogel et al. (2021) deposited Fastq files in the National Center of Biotechnology Information Sequence Read Archive (BioProject accession number PRJNA616021) and provided the scripts for analysis via a GitHub repository (<https://github.com/gmv23/pcap-gwas>), but we were unable to find a DOI that refers to these materials or a citation to properly cite them here and this illustrates that simply depositing code and data in repositories is not enough. Efforts must be made to annotate and structure the raw data to make it FAIR (Wilkinson et al. 2016). Making the raw data FAIR means providing proper metadata, a clear description of what the resource contains, how it can be used and other attributes about it that help users understand it. Further, fully documenting the code used for the analysis is necessary for other researchers to fully use and understand.

The third layer is the use of proper code repositories such as a library resource, or code repository (e.g., GitHub or GitLab) and data repositories (e.g., FigShare, Zenodo or OSF.io), allowing for the deposition and updating of code, figures, data preprints or any other materials that support the article itself while providing a DOI and citation. As an example, Sparks et al. (2011, 2014; 2016) used FigShare to provide models, data and code necessary to replicate model development and the subsequent study on the effects of climate change on potato late blight. Similarly, Carleson et al. (2019) hosted the code for reproducing a population genomic analysis of *Phytophthora plurivora* on GitHub, while providing all data on OSF.io (the Open Science Framework). Lehner et al. (2017) used GitHub to host a code repository of their research compendium website with data and a reproducible report that explains in detail all steps of the analysis and the R code for conducting a meta-analysis for assessing heterogeneity in the relationship between white mold incidence and soybean yield and between incidence and soybean yield (<https://emdelponte.github.io/paper-white-mold-meta-analysis/>). The website clearly demonstrates the analysis to readers and uses R so that anyone can easily replicate the work. Using a public code repository resource allows other researchers to easily contact the authors by opening “Issues” and report bugs or ask questions in an open forum that are not as straightforward when the data are provided as supplementary material.

Taking this approach even further, packaging the full analysis in a containerized software application (Docker) is a way to help ensure computational reproducibility but at the expense of added complexity (Merkel 2014). Docker is an open-source containerization platform that enables users to package several applications and an operating system into containers, thereby standardizing the executable components by combining application source code (or analyses) with the operating system required to run that analysis on a user’s computer. However, there are drawbacks to using Docker. It can be difficult to understand for a new user and new platforms, like the Apple M1 chip, may not be fully supported, which hinder the ability to share the container. But, in most cases using an open-source language like R, Julia or Python allows us to share our work in a fashion where we know that the analysis will run exactly the same on every computer. For more on using Docker for reproducible research, we would refer readers to Nüst et al. (2020). As an example of this approach, Khaliq et al. (2020a) provided a research compendium as a Docker container with a DOI and a full R package. This enables readers to fully replicate their analysis of *Ascochyta rabiei* conidia dispersal in chickpea using the data collected. It also allows for more in-depth investigation by stepping through other points where weather data were investigated and various models were fit before deciding on the best fit and recreate any figures as published in the article (Khaliq et al. 2020b). When this layer is employed, tools such as Binder, <https://mybinder.org>, can be used that allow readers and reviewers to launch an interactive session in their web browser and interact with the data and rerun the analysis in an RStudio instance or Jupyter notebooks, a “web application for creating and sharing computational documents” (Jupyter 2022), as Mioroni et al. used (2018, 2019). Kamvar et al. (2015) took a slightly different approach by including all files necessary for the analysis and most output files in a repository (Kamvar et al. 2014) that also included an installable R package that was used for the original analysis. While there are many other methods, these two approaches illustrate some of the best-practices where the data and other files were deposited in repositories with DOIs and reproducibility issues were addressed by using R packages to handle dependencies and other versioning issues making the work more portable.

These layers provide increasing openness and reproducibility so Layer 1 should not be viewed as the starting point before moving on to Layer 2 and then Layer 3. Rather the layers are provided to give the reader some idea of how much benefit end-users of the computational materials and data will derive at the expense of the increasing complexity. Authors need not start with Level 1 before moving to Level 2 and then Level 3. If they have the ability to start with depositing the computational materials and data and generating DOIs all parties will benefit from it. However, we recognize that these steps may not be appropriate or attractive for various reasons and so have detailed other layers that improve openness and reproducibility.

In this work we have evaluated the state of computational reproducibility in the plant pathology literature and presented suggestions for areas of improvement. As we prepared this letter, we became more aware of the urgent need to spread and establish an open science attitude and culture among plant pathologists. To assist in fostering this sort of change in our discipline, Open Plant Pathology (OPP), <https://www.openplantpathology.org>, an institution-independent and non-funded initiative, was founded in January 2018 by two of this letter’s co-authors, Del Ponte and Sparks, in 2018 with the following vision: “foster a diverse community culture that values openness, transparency and reproducibility of scientific research data and methods in our field”. We started OPP with a minimal infrastructure and support from other enthusiastic colleagues that allows members to interact sharing and gathering ideas on how we can improve the openness and reproducibility in our discipline.

We believe that adopting an attitude of open and collaborative science and using the best reproducibility practices in our daily work, directly benefits us as researchers. For example, between complicated analyses, reviews and revisions and questions years later about the data that was collected or analysis that was conducted, it is extremely beneficial to be able to easily reproduce our work quickly and easily. This manuscript was drafted over the course of several years as the authors had time to devote to it. Having everything in a reproducible framework made it easy to resume work and set-aside as necessary without losing information and having everything well-documented made it simpler to do this. Second, it benefits the reviewer by aiding their understanding of the work done and gives them more materials to use to make suggestions for improvements when reviewing and the end-user or reader is better able to verify the validity of the methods used and recreate the analysis. Perhaps more importantly, sharing these details helps with knowledge transfer by showing other interested parties how something was done rather than simply describing it. Lastly, openly sharing our work and making it discoverable can lead to new collaborations and synergistic ideas. One of the most important messages that we would like to share is that there is more to the work than just the paper. Sharing materials detailing the analysis that was performed and documenting the data provide citable products and enhance the manuscript providing the reader with a richer set of information with which to understand the work that was performed. This open sharing of code and data leads to greater impact as work is cited if resources such as code or data are reused. But it is not just up to the authors to ensure that their work is reproducible. At the very least, journals can and should provide clear instruction for how to deposit the data and code in a repository and mint a DOI to accompany these resources to encourage authors to share the data and code that support the manuscript. Ideally, with mandates for openly sharing data becoming more common with funding agencies, journals should also be mandating this practice as well. We can and should embrace this and move the discipline forward and have a greater impact with our work.

# Authors’ Contributions

A.H.S, E.DP., Z.F. and N.J.G. conceived of the presented idea and scoring system. E.DP. provided five-year impact factors. A.H.S., E.DP., Z.F. K.A. and N.G. evaluated articles for scoring. A.H.S. designed the computational framework for analysis and created the research compendium with feedback from E.DP. and Z.F. A.H.S. wrote the manuscript in consultation with E.DP. All authors provided critical feedback and helped contribute to the final version of the manuscript.

# Technical Details

## Data

The raw data for this work are documented and available from DOI [https://doi.org/10.5281/zenodo.4941722](10.5281/zenodo.4941722).

## Code

All code used in the analyses and data visualization and associated materials have been made available as research compendium available from DOI [https://doi.org/10.5281/zenodo.1250664](10.5281/zenodo.1250664). A webpage version of the compendium is available from <https://openplantpathology.github.io/Reproducibility_in_Plant_Pathology/>.

## Computational Details

All relevant computational information (R and package versions, operating system) are given in the Methods for the data gathering and analysis. The relevant details for this article itself are shown here.

**R version 4.2.1 (2022-06-23)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_AU.UTF-8||en\_AU.UTF-8||en\_AU.UTF-8||C||en\_AU.UTF-8||en\_AU.UTF-8

**attached base packages:** *tools*, *stats*, *graphics*, *grDevices*, *utils*, *datasets*, *methods* and *base*

**other attached packages:** *Reproducibility.in.Plant.Pathology(v.0.0.7)*, *tidyr(v.1.2.1)*, *rsvg(v.2.3.1)*, *rope(v.1.0)*, *report(v.0.4.0)*, *posterior(v.1.3.1)*, *patchwork(v.1.1.2)*, *parameters(v.0.19.0)*, *officer(v.0.4.4)*, *knitr(v.1.40)*, *janitor(v.2.1.0)*, *here(v.1.0.1)*, *ggpubr(v.0.4.0)*, *ggplot2(v.3.3.6)*, *flextable(v.0.8.2)*, *extrafont(v.0.18)*, *effectsize(v.0.8.1)*, *english(v.1.2-6)*, *dplyr(v.1.0.10)*, *cowplot(v.1.1.1)*, *brms(v.2.18.0)*, *Rcpp(v.1.0.9)*, *bayestestR(v.0.13.0)*, *bayesplot(v.1.9.0)*, *DiagrammeRsvg(v.0.1)* and *DiagrammeR(v.1.0.9)*

**loaded via a namespace (and not attached):** *uuid(v.1.1-0)*, *backports(v.1.4.1)*, *systemfonts(v.1.0.4)*, *plyr(v.1.8.7)*, *igraph(v.1.3.5)*, *splines(v.4.2.1)*, *crosstalk(v.1.2.0)*, *TH.data(v.1.1-1)*, *rstantools(v.2.2.0)*, *inline(v.0.3.19)*, *digest(v.0.6.29)*, *htmltools(v.0.5.3)*, *fansi(v.1.0.3)*, *magrittr(v.2.0.3)*, *checkmate(v.2.1.0)*, *tzdb(v.0.3.0)*, *readr(v.2.1.3)*, *RcppParallel(v.5.1.5)*, *matrixStats(v.0.62.0)*, *xts(v.0.12.1)*, *sandwich(v.3.0-2)*, *extrafontdb(v.1.0)*, *prettyunits(v.1.1.1)*, *colorspace(v.2.0-3)*, *xfun(v.0.33)*, *callr(v.3.7.2)*, *crayon(v.1.5.2)*, *jsonlite(v.1.8.2)*, *lme4(v.1.1-30)*, *survival(v.3.4-0)*, *zoo(v.1.8-11)*, *glue(v.1.6.2)*, *gtable(v.0.3.1)*, *emmeans(v.1.8.1-1)*, *V8(v.4.2.1)*, *distributional(v.0.3.1)*, *car(v.3.1-0)*, *pkgbuild(v.1.3.1)*, *Rttf2pt1(v.1.3.11)*, *rstan(v.2.21.7)*, *abind(v.1.4-5)*, *scales(v.1.2.1)*, *mvtnorm(v.1.1-3)*, *DBI(v.1.1.3)*, *rstatix(v.0.7.0)*, *miniUI(v.0.1.1.1)*, *xtable(v.1.8-4)*, *stats4(v.4.2.1)*, *StanHeaders(v.2.21.0-7)*, *DT(v.0.25)*, *datawizard(v.0.6.2)*, *htmlwidgets(v.1.5.4)*, *threejs(v.0.3.3)*, *RColorBrewer(v.1.1-3)*, *ellipsis(v.0.3.2)*, *pkgconfig(v.2.0.3)*, *loo(v.2.5.1)*, *farver(v.2.1.1)*, *utf8(v.1.2.2)*, *tidyselect(v.1.2.0)*, *rlang(v.1.0.6)*, *reshape2(v.1.4.4)*, *later(v.1.3.0)*, *munsell(v.0.5.0)*, *visNetwork(v.2.1.2)*, *cli(v.3.4.1)*, *generics(v.0.1.3)*, *broom(v.1.0.1)*, *ggridges(v.0.5.4)*, *evaluate(v.0.17)*, *stringr(v.1.4.1)*, *fastmap(v.1.1.0)*, *yaml(v.2.3.5)*, *processx(v.3.7.0)*, *pander(v.0.6.5)*, *zip(v.2.2.1)*, *purrr(v.0.3.5)*, *nlme(v.3.1-159)*, *mime(v.0.12)*, *projpred(v.2.2.1)*, *xml2(v.1.3.3)*, *compiler(v.4.2.1)*, *shinythemes(v.1.2.0)*, *rstudioapi(v.0.14)*, *curl(v.4.3.3)*, *gamm4(v.0.2-6)*, *ggsignif(v.0.6.3)*, *tibble(v.3.1.8)*, *stringi(v.1.7.8)*, *ps(v.1.7.1)*, *Brobdingnag(v.1.2-7)*, *gdtools(v.0.2.4)*, *readODS(v.1.7.0)*, *lattice(v.0.20-45)*, *Matrix(v.1.5-1)*, *nloptr(v.2.0.3)*, *markdown(v.1.1)*, *shinyjs(v.2.1.0)*, *tensorA(v.0.36.2)*, *vctrs(v.0.4.2)*, *pillar(v.1.8.1)*, *lifecycle(v.1.0.3)*, *bridgesampling(v.1.1-2)*, *estimability(v.1.4.1)*, *data.table(v.1.14.4)*, *insight(v.0.18.5)*, *httpuv(v.1.6.6)*, *R6(v.2.5.1)*, *bookdown(v.0.29)*, *promises(v.1.2.0.1)*, *gridExtra(v.2.3)*, *codetools(v.0.2-18)*, *boot(v.1.3-28)*, *colourpicker(v.1.1.1)*, *MASS(v.7.3-58.1)*, *gtools(v.3.9.3)*, *assertthat(v.0.2.1)*, *rprojroot(v.2.0.3)*, *withr(v.2.5.0)*, *shinystan(v.2.6.0)*, *multcomp(v.1.4-20)*, *hms(v.1.1.2)*, *mgcv(v.1.8-40)*, *parallel(v.4.2.1)*, *grid(v.4.2.1)*, *coda(v.0.19-4)*, *minqa(v.1.2.4)*, *snakecase(v.0.11.0)*, *rmarkdown(v.2.17)*, *carData(v.3.0-5)*, *lubridate(v.1.8.0)*, *shiny(v.1.7.2)*, *base64enc(v.0.1-3)* and *dygraphs(v.1.1.1.6)*

# Acknowledgements

The authors are grateful for insightful comments from Dr David Ferris and Dr Rebecca O’Leary the USQ Centre for Crop Health Advisory Group on the final manuscript that helped improve the paper and from Ms Anna Hepworth for statistical guidance on evaluating inter-rater scores and for the comments from the anonymous reviewers and Dr Nian Wang that greatly improved the quality of this manuscript.

# Literature Cited

Arnaud, E., Laporte, M.-A., Kim, S., Aubert, C., Leonelli, S., Miro, B., et al. 2020. The ontologies community of practice: A CGIAR initiative for big data in agrifood systems. Patterns. 1:100105 Available at: <https://www.sciencedirect.com/science/article/pii/S2666389920301392>.

Australian Research Council. 2018. ARC Open Access Policy. Available at: <https://www.arc.gov.au/policies-strategies/policy/arc-open-access-policy> [Accessed April 20, 2020].

Baker, M. 2016. [Muddled meanings hamper efforts to fix reproducibility crisis](https://doi.org/10.1038/nature.2016.20076). Nature News.

Barton, C. M., Lee, A., Janssen, M. A., Leeuw, S. van der, Tucker, G. E., Porter, C., et al. 2022. How to make models more useful. Proceedings of the National Academy of Sciences. 119:e2202112119 Available at: <https://www.pnas.org/doi/abs/10.1073/pnas.2202112119>.

Brunsdon, C. 2015. [Quantitative methods I: Reproducible research and quantitative geography](https://doi.org/10.1177/0309132515599625). Prog. Hum. Geogr.

Bürkner, P.-C. 2018. [Advanced Bayesian multilevel modeling with the R package brms](https://doi.org/10.32614/RJ-2018-017). The R Journal. 10:395–411.

Bürkner, P.-C. 2021. [Bayesian item response modeling in R with brms and Stan](https://doi.org/10.18637/jss.v100.i05). Journal of Statistical Software. 100:1–54.

Bürkner, P.-C. 2017. [brms: An R package for Bayesian multilevel models using Stan](https://doi.org/10.18637/jss.v080.i01). Journal of Statistical Software. 80:1–28.

Carleson, N. C., Fieland, V. J., Scagel, C. F., Weiland, J. E., and Grünwald, N. J. 2019. Population structure of *Phytophthora plurivora* on rhododendron in Oregon nurseries. Plant disease. 103:1923–1930.

Del Ponte, E. M. 2020. [Towards a more open and transparent plant pathology research](https://doi.org/10.1007/s40858-020-00388-z). Tropical Plant Pathology. 45:361–362.

Del Ponte, E. M., Nelson, S. C., and Pethybridge, S. J. 2019. [Evaluation of app-embedded disease scales for aiding visual severity estimation of cercospora leaf spot of table beet](https://doi.org/10.1094/pdis-10-18-1718-re). Plant Disease. 103:1347–1356.

Dienlin, T., Johannes, N., Bowman, N. D., Masur, P. K., Engesser, S., Kümpel, A. S., et al. 2020. [An agenda for open science in communication](https://doi.org/10.1093/joc/jqz052). Journal of Communication. 71:1–26.

Eckert, E. M., Cesare, A. D., Fontaneto, D., Berendonk, T. U., Bürgmann, H., Cytryn, E., et al. 2020. [Every fifth published metagenome is not available to science](https://doi.org/10.1371/journal.pbio.3000698). PLOS Biology. 18:e3000698.

Editors at Nature. 2016. [Reality check on reproducibility](https://doi.org/10.1038/533437a). Nature. 533:437.

European Commission. 2022. Open science. Available at: <https://research-and-innovation.ec.europa.eu/strategy/strategy-2020-2024/our-digital-future/open-science_en>.

Fernandez, J. 2022. [jafernandez01/cndcR: v1.0.0](https://doi.org/10.5281/ZENODO.6468142).

Fernandez, J. A., Versendaal, E. van, Lacasa, J., Makowski, D., Lemaire, G., and Ciampitti, I. A. 2022a. Dataset characteristics for the determination of critical nitrogen dilution curves: From past to new guidelines. European Journal of Agronomy. 139:126568 Available at: <https://www.sciencedirect.com/science/article/pii/S1161030122001162>.

Fernandez, J. A., Versendaal, E. van, Lacasa, J., Makowski, D., Lemaire, G., and Ciampitti, I. A. 2022b. Dataset characteristics for the determination of critical nitrogen dilution curves: From the past to new guidelines. European Journal of Agronomy. in revision.

Fidler, F., and Gordon, A. 2013. Science is in a reproducibility crisis: How do we resolve it? Phys.org. Available at: <https://phys.org/news/2013-09-science-crisis.html>.

Fitzjohn, R., Pennell, M., Zanne, A., and Cornwell, W. 2014. Reproducible research is still a challenge. ROpensci.org. 2014 Available at: <http://ropensci.org/blog/2014/06/09/reproducibility/>.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., and Gelman, A. 2019. [Visualization in Bayesian workflow](https://doi.org/10.1111/rssa.12378). J. R. Stat. Soc. A. 182:389–402.

Gates Foundation. 2022. Gates foundation open access policy. Available at: <https://openaccess.gatesfoundation.org/open-access-policy/>.

Government of Canada. 2016. Tri-Agency Open Access Policy on Publications - Science.gc.ca. Tri-Agency Open Access Policy on Publications. Available at: <http://www.science.gc.ca/eic/site/063.nsf/eng/h_F6765465.html?OpenDocument> [Accessed April 12, 2019].

Hill, G. N., Beresford, R. M., and Evans, K. J. 2019. [Automated analysis of aggregated datasets to identify climatic predictors of botrytis bunch rot in wine grapes](https://doi.org/10.1094/phyto-10-17-0357-r). Phytopathology. 109:84–95.

Ioannidis, J. P. A. 2014. [How to make more published research true](https://doi.org/10.1371/journal.pmed.1001747). PLoS Medicine. 11.

Iqbal, S. A., Wallach, J. D., Khoury, M. J., Schully, S. D., and Ioannidis, J. P. A. 2016. [Reproducible research practices and transparency across the biomedical literature](https://doi.org/10.1371/journal.pbio.1002333). PLoS Biology. 14:1–13.

Jupyter, P. 2022. Jupyter. Available at: <https://jupyter.org>.

Kamvar, Z. N., Larsen, M. M., Kanaskie, A. M., Hansen, E. M., and Grünwald, N. J. 2015. [Spatial and temporal analysis of populations of the sudden oak death pathogen in Oregon forests](https://doi.org/10.1094/PHYTO-12-14-0350-FI). Phytopathology. 105:982–989.

Kamvar, Z. N., Larsen, M. M., Kanaskie, A. M., Hansen, E. M., and Grünwald, N. J. 2014. [Sudden Oak Death in Oregon Forests: Spatial and temporal population dynamics of the sudden oak death epidemic in Oregon Forests](https://doi.org/10.5281/zenodo.13007).

Khaliq, I., Fanning, J., Melloy, P., Galloway, J., Moore, K., Burrell, D., et al. 2020a. [The role of conidia in the dispersal of *Ascochyta rabiei*](https://doi.org/10.1007/s10658-020-02126-2). European Journal of Plant Pathology. 158:911–924.

Khaliq, I., Melloy, P., Burrell, D., Fanning, J., Galloway, J., Moore, K., et al. 2020b. *ChickpeaAscoDispersal: A research compendium to accompany “the role of conidia in the dispersal of Ascochyta rabiei”*. Available at: <https://doi.org/10.5281/zenodo.3810826>.

Kool, H., Andersson, J. A., and Giller, K. E. 2020. [Reproducibility and external validity of on-farm experimental research in Africa](https://doi.org/10.1017/s0014479720000174). Experimental Agriculture. 56:587–607.

Lehner, M. S., Pethybridge, S. J., Meyer, M. C., and Del Ponte, E. M. 2017. [Meta-analytic modelling of the incidence–yield and incidence–sclerotial production relationships in soybean white mould epidemics](https://doi.org/10.1111/ppa.12590). Plant Pathology. 66:460–468.

Madden, L. V., Shah, D. A., and Esker, P. D. 2015. [Does the *p* value have a future in plant pathology?](https://doi.org/10.1094/PHYTO-07-15-0165-LE) Phytopathology. 105:1400–1407.

Makowski, D., Ben-Shachar, M. S., and Lüdecke, D. 2019. [bayestestR: Describing effects and their uncertainty, existence and significance within the Bayesian framework](https://doi.org/10.21105/joss.01541). Journal of Open Source Software. 4:1541.

Makowski, D., Ben-Shachar, M. S., Patil, I., and Lüdecke, D. 2021. Automated results reporting as a practical tool to improve reproducibility and methodological best practices adoption. CRAN. Available at: <https://github.com/easystats/report>.

Merkel, D. 2014. Docker: Lightweight Linux containers for consistent development and deployment. Linux Journal. 2014:2.

Miorini, T. J. J., Kamvar, Z. N., Higgins, R. S., Raetano, C. G., Steadman, J. R., and Everhart, S. E. 2019. [Differential aggressiveness of *Sclerotinia sclerotiorum* isolates from North and South America and partial host resistance in Brazilian soybean and dry bean cultivars](https://doi.org/10.1007/s40858-018-00273-w). Tropical Plant Pathology. 44:73–81.

Miorini, T. J. J., Kamvar, Z. N., Higgins, R., Raetano, C. G., Steadman, J., and Everhart, S. E. 2018. Data and analysis for variation in pathogen aggression and cultivar performance against *Sclerotinia sclerotiorum* in soybean and dry bean from the U.S. And Brazil. Available at: <https://osf.io/2x7fc/> [Accessed July 29, 2020].

Noorden, R. van. 2017. Gates Foundation demands open access. Nature. 541:270–270.

Nüst, D., Sochat, V., Marwick, B., Eglen, S. J., Head, T., Hirst, T., et al. 2020. [Ten simple rules for writing dockerfiles for reproducible data science](https://doi.org/10.1371/journal.pcbi.1008316) ed. Scott Markel. PLOS Computational Biology. 16:e1008316.

Ooi, H., Vries, A. de, and Microsoft. 2022. *checkpoint: Install packages from snapshots on the Checkpoint server for reproducibility*. Available at: <https://CRAN.R-project.org/package=checkpoint>.

Patil, P., Peng, R. D., and Leek, J. 2016. A statistical definition for reproducibility and replicability. bioRxiv. Available at: <http://biorxiv.org/content/early/2016/07/29/066803>.

Peng, R. D. 2009. [Reproducible research and biostatistics](https://doi.org/10.1093/biostatistics/kxp014). Biostatistics. 10:405–408.

Preeyanon, L., Pyrkosz, A. B., and Titus Brown, C. 2018. [Reproducible bioinformatics research for biologists](https://doi.org/10.1201/9781315373461-7). In Chapman; Hall/CRC, p. 185–217.

R Core Team. 2022. *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Available at: <https://www.R-project.org/>.

Science, O. W., and Oliveira, M. C. 2021. *Maxwelco/OpenWeedSci: First release*. Zenodo. Available at: <https://doi.org/10.5281/zenodo.5079863>.

Seibold, H., Czerny, S., Decke, S., Dieterle, R., Eder, T., Fohr, S., et al. 2021. [A computational reproducibility study of PLOS ONE articles featuring longitudinal data analyses](https://doi.org/10.1371/journal.pone.0251194) ed. Jelte M. Wicherts. PLOS ONE. 16:e0251194.

Serra da Cruz, S. M., and Pires do Nascimento, J. A. 2019. Towards integration of data-driven agronomic experiments with data provenance. Computers and Electronics in Agriculture. 161:14–28 Available at: <https://www.sciencedirect.com/science/article/pii/S0168169917315004>.

Smith, F., Dodds, L., Day, C., Musker, R., and Parr, M. 2018. [Creating FAIR and open data ecosystems for agricultural programmes](https://doi.org/10.21955/GATESOPENRES.1114883.1).

Sparks, A. 2016. Global-late-blight (meta)modelling. Available at: <https://figshare.com/articles/dataset/Global_Late_Blight_Modelling/1066124/6>.

Sparks, A. H., Del Ponte, E. M., Alves, K. S., Foster, Z., and Grünwald, N. J. 2022. Compendium of r code and data for ‘status and best practices for reproducible research in plant pathology. Available at: <https://doi.org/10.5281/zenodo.1250665>.

Sparks, A. H., Forbes, G. A., Hijmans, R. J., and Garrett, K. A. 2011. [A metamodeling framework for extending the application domain of process-based ecological models](https://doi.org/10.1890/ES11-00128.1). Ecosphere. 2:art90.

Sparks, A. H., Forbes, G. A., Hijmans, R. J., and Garrett, K. A. 2014. [Climate change may have limited effect on global risk of potato late blight](https://doi.org/10.1094/PDIS-04-11-031). Global Change Biology. :3621–3631.

Stacey, D., Wulff, K., Chikhalla, N., and Bernardo, T. 2022. [From FAIR to FAIRS: Data security by design for the global burden of animal diseases](https://doi.org/10.1002/agj2.21017). Agronomy Journal. 114:2693–2699.

Stodden, V., Guo, P., and Ma, Z. 2013. [Toward reproducible computational research: An empirical analysis of data and code policy adoption by journals](https://doi.org/10.1371/journal.pone.0067111). PLoS One. 8:2–9.

Sweedler, J. V. 2015. [Striving for reproducible science](https://doi.org/10.1021/acs.analchem.5b04300). Analytical Chemistry. 87:11603–11604.

Tiwari, K., Kananathan, S., Roberts, M. G., Meyer, J. P., Shohan, M. U. S., Xavier, A., et al. 2021. [Reproducibility in systems biology modelling](https://doi.org/10.15252/msb.20209982). Molecular Systems Biology. 17.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. 2021. [Rank-normalization, folding, and localization: An improved Rˆ for assessing convergence of MCMC (with discussion)](https://doi.org/10.1214/20-ba1221). Bayesian Analysis. 16.

Vogel, G., Gore, M. A., and Smart, C. D. 2021. [Genome-wide association study in new york iphytophthora capsici/i isolates reveals loci involved in mating type and mefenoxam sensitivity](https://doi.org/10.1094/phyto-04-20-0112-fi). Phytopathology. 111:204–216.

Wald, C. 2010. Scientists embrace openness. Available at: <https://www.science.org/content/article/scientists-embrace-openness> [Accessed May 1, 2022].

Wallach, J. D., Boyack, K. W., and Ioannidis, J. P. A. 2018. [Reproducible research practices, transparency, and open access data in the biomedical literature, 20152017](https://doi.org/10.1371/journal.pbio.2006930) ed. Ulrich Dirnagl. PLOS Biology. 16:e2006930.

Weissgerber, T. L., Garovic, V. D., Winham, S. J., Milic, N. M., and Prager, E. M. 2016. [Transparent reporting for reproducible science](https://doi.org/10.1002/jnr.23785). Journal of Neuroscience Research. 00:1–6.

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data. 3:1–9 Available at: <https://www.nature.com/articles/sdata201618> [Accessed April 20, 2020].

Wittman, J. T., and Aukema, B. H. 2020. A guide and toolbox to replicability and open science in entomology. Journal of Insect Science. 20 Available at: <https://doi.org/10.1093/jisesa/ieaa036>.

Ziemann, M., Eren, Y., and El-Osta, A. 2016. [Gene name errors are widespread in the scientific literature](https://doi.org/10.1186/s13059-016-1044-7). Genome Biology. 17:177.

# Tables

Table 1: Journal titles selected for inclusion representing 21 plant pathology journals or specialized journals featuring plant pathology focused articles, their respective five-year impact factors as of 2022 (5 Year IF), number of articles per year and the total number (n) of articles from each journal that were evaluated.

| Journal | IF | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | n |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Australasian Plant Pathology | 1.708 | 0 | 2 | 2 | 0 | 0 | 3 | 1 | 0 | 2 | 1 | 11 |
| Canadian Journal of Plant Pathology | 1.993 | 3 | 1 | 1 | 2 | 0 | 3 | 3 | 3 | 2 | 1 | 19 |
| Crop Protection | 3.110 | 3 | 2 | 2 | 1 | 1 | 4 | 1 | 1 | 3 | 5 | 23 |
| European Journal of Plant Pathology | 2.022 | 1 | 4 | 1 | 1 | 1 | 2 | 1 | 5 | 1 | 2 | 19 |
| Forest Pathology | 1.595 | 3 | 1 | 2 | 1 | 2 | 3 | 2 | 2 | 1 | 1 | 18 |
| Journal of Phytopathology | 1.574 | 0 | 4 | 1 | 2 | 0 | 2 | 5 | 4 | 1 | 0 | 19 |
| Journal of Plant Pathology | 1.681 | 3 | 3 | 1 | 1 | 3 | 1 | 3 | 4 | 4 | 0 | 23 |
| Journal of General Plant Pathology | 1.416 | 4 | 3 | 2 | 1 | 2 | 3 | 2 | 1 | 3 | 3 | 24 |
| Molecular Plant-Microbe Interactions | 4.836 | 2 | 2 | 2 | 0 | 4 | 2 | 4 | 2 | 0 | 4 | 22 |
| Molecular Plant Pathology | 5.626 | 6 | 3 | 2 | 1 | 3 | 2 | 1 | 3 | 5 | 3 | 29 |
| Nematology | 1.485 | 0 | 2 | 3 | 0 | 2 | 1 | 3 | 2 | 1 | 2 | 16 |
| Physiological and Molecular Plant Pathology | 2.388 | 4 | 1 | 3 | 0 | 3 | 3 | 4 | 0 | 0 | 6 | 24 |
| Phytoparasitica | 1.569 | 2 | 5 | 0 | 4 | 2 | 1 | 5 | 2 | 1 | 1 | 23 |
| Phytopathologia Mediterranea | 2.080 | 2 | 1 | 1 | 0 | 3 | 2 | 0 | 0 | 4 | 7 | 20 |
| Phytopathology | 4.394 | 4 | 2 | 3 | 4 | 3 | 2 | 1 | 5 | 3 | 1 | 28 |
| Plant Disease | 4.700 | 1 | 3 | 2 | 3 | 0 | 3 | 3 | 3 | 4 | 2 | 24 |
| Plant Health Progress | 0.000 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 4 | 2 | 0 | 16 |
| Plant Pathology | 2.924 | 2 | 3 | 4 | 1 | 4 | 5 | 5 | 2 | 3 | 1 | 30 |
| Revista Mexicana de Fitopatología | 0.000 | 4 | 1 | 1 | 2 | 0 | 0 | 2 | 1 | 6 | 2 | 19 |
| Tropical Plant Pathology | 1.675 | 1 | 1 | 3 | 1 | 0 | 2 | 1 | 4 | 2 | 3 | 18 |
| Virology Journal (Plant Viruses Section) | 3.719 | 0 | 3 | 4 | 2 | 2 | 4 | 1 | 3 | 2 | 4 | 25 |

Table 2: The frequency of use of top ten most used software programs that were found in 450 papers published in 21 plant pathology journals or plant pathology focused articles from other specialized journals by year of publication.

| Software | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | Total |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MEGA | 4 | 5 | 5 | 5 | 7 | 9 | 8 | 14 | 9 | 7 | 73 |
| SAS | 9 | 9 | 9 | 9 | 5 | 7 | 5 | 5 | 7 | 5 | 70 |
| SPSS | 3 | 3 | 5 | 1 | 4 | 6 | 5 | 8 | 8 | 5 | 48 |
| R | 1 | 1 | 3 | 2 | 5 | 4 | 3 | 10 | 7 | 5 | 41 |
| CLUSTAL | 3 | 1 | 2 | 0 | 3 | 8 | 3 | 5 | 1 | 2 | 28 |
| BLAST | 5 | 5 | 3 | 1 | 2 | 0 | 0 | 1 | 1 | 4 | 22 |
| MRBAYES | 0 | 1 | 0 | 1 | 0 | 1 | 4 | 5 | 2 | 2 | 16 |
| GENSTAT | 3 | 1 | 1 | 1 | 2 | 0 | 2 | 2 | 1 | 1 | 14 |
| STATISTICA | 1 | 3 | 4 | 0 | 1 | 2 | 0 | 1 | 1 | 1 | 14 |
| EXCEL | 3 | 1 | 1 | 1 | 1 | 2 | 0 | 2 | 2 | 0 | 13 |
| BIOEDIT | 0 | 0 | 1 | 0 | 1 | 3 | 0 | 4 | 0 | 1 | 10 |

# Figures