Ten Simple Rules for Digital Data Storage

Edmund Hart [1,*], Pauline Barmby [2], David LeBauer [3], Naupaka Zimmerman [4], Kara H. Woo [5], Sarah Mount [6], Timothée Poisot [7], François Michonneau [8,9], Jeffrey Hollister [10],

1 Univeristy of Vermont, Department of Biology, Burlington
2 University of Western Ontario, Department of Physics and Astronomy
3 University of Illinois at Urbana-Champaign, National Center for Supercomputing Applications and Institute for Genomic Biology
4 University of Arizona, School of Plant Sciences
5 Washington State University, Center for Environmental Research, Education, and Outreach
6 University of Wolverhampton, School of Mathematics and Computer Science
7 Université de Montréal, Département de Sciences Biologiques
8 University of Florida, iDigBio, Florida Museum of Natural History
9 University of Florida, Whitney Laboratory for Marine Biosciences
10 US Environmental Protection Agency, Atlantic Ecology Division

∗ **E-mail: edmund.m.hart@gmail.com**

Introduction

Data is the central currency of science, but the nature of scientific data has changed dramatically with the rapid pace of technological change. This change led to an increasing heterogeneity of data formats, dataset sizes, data complexity, data use cases, and data sharing. For example, improvements in high throughput DNA sequencing, sustained institutional support for large sensor networks [1,2], and sky surveys with large-format digital cameras [3] created massive quantities of data. At the same time, increasingly common collaboration between researchers [4] and data aggregation in portals (e.g. for biodiversity data, GBIF or iDigBio) necessitates increased coordination among data collectors and institutions [5,6]. As a consequence, "data" can now mean anything from petabytes of information stored in professionally-maintained databases, through spreadsheets on a single computer, to hand-written tables in lab notebooks on shelves. All remain important, but methods of data curation must continue be updated in order to encompass the changes brought about by new forms and practices of data collection and storage.

While much has been written about both the virtues of data sharing [7,8] and best practices to do so [9,10], how to store data has received comparatively less attention. Proper storage is a prerequisite to sharing, and indeed inadequate storage contributes to the phenomenon of data decay or "data entropy": data, whether publicy shared or not, becomes less accessible through time [11–14]. Best practices for data storage often begin and end with, "use a community standard repository." This is, by all means, a great practice; however, data storage policies are highly variable between repositories [15], and best practices across all stages of the data life cycle will facilitate transition from local storage to repository. Good storage practices are important even (or especially) in cases where data may not fit with an existing repository, in the cases

where only derived data products (versus raw data) are suitable for deposition, or in the case where an existing repository may have lax standards.

Therefore, this manuscript describes 10 simple rules for digital data storage that grew out of a long discussion among instructors for the Software and Data Carpentry initiative [16,17]. Software and Data Carpentry instructors are scientists from diverse backgrounds who have encountered a variety of data storage challenges and are active in teaching other scientists best practices for scientific computing and data management. Thus, this paper represents a distillation of collective experience, and hopefully will be useful to scientists facing a variety of data storage challenges.

Rule 1: Know what to expect

One can avoid most of the troubles encountered during the analysis, management, and release of data by having a clear roadmap of what to expect *before* data acquisition starts. For instance:

- How will the raw data be received? Are they delivered by a machine or software, or typed-in?
- What is the format expected by the software used for analysis?
- Is there a community standard format?

The answers to these questions can range from simple cases (e.g., sequencing data stored in the FASTA format, which can be used "as is" throughout the analysis), to experimental designs involving multiple instruments, each with its own output format and conventions. Knowing the state in which the data needs to be at each step of the analysis can help (i) identify software tools to use in converting across data formats, (ii) orient technological choices about how and where the data should be stored, and (iii) rationalize the analysis pipeline, making it more amenable to re-use.

Also key is the ability to estimate the storage volume needed to store the data, both during and after the analysis. The required strategy will differ for datasets of varying size. Smaller datasets (e.g. a few megabytes in size) can be managed locally with a simple data management plan, whereas larger datasets (e.g. gigabytes to petabytes) will in almost all cases require careful planning and preparation (Rule 9).

Rule 2: Know your use case

Well identified use cases make data storage easier. Ideally prior to beginning data collection, one can answer the following questions:

- Should the raw data be archived (Rule 3)?
- Should the data used for analysis be prepared once, or re-generated from the raw data each time (and what difference would this choice make for storage, computing requirements, and reproducibility)?
- Can manual corrections be avoided in favor of programmatic approaches?
- How will changes to the data be tracked, and where will these tracked changes be logged?
- Will the final data be released, and if so, in what format?
- Are there restrictions or privacy concerns associated with the data (e.g. for survey results with personally identifiable information (PII), threatened species, or confidential business information?
- Will institutional validation be required prior to releasing the data?

- Does the funding agency mandate data deposition in a publicly available archive, and if so, where and under what license?
- Does the target journal mandate data deposition?

None of these questions have universal answers, nor are they the only questions one should ask before starting data acquisition. But, as for Rule 1, knowing the what, when, and how of *your* use of the data will bring you close to a reliable roadmap on how to handle data from acquisition through publication to archive.

Rule 3: Keep raw data raw

Since analytical and data processing procedures improve or otherwise change over time, having access to the 'raw' (unprocessed) data can facilitate future re-analysis and analytical reproducibility. As processing algorithms improve and computational power increase, new analyses will be enabled that were not possible at the time of the original work. If only derived data are stored, it can be difficult to impossible for other researchers to confirm analytical results, to assess the validity of statistical models, or to directly compare findings across studies.

Therefore, data should always be kept in raw format whenever possible (within the constraints of technical limitations). In addition to being the most appropriate way to ensure transparency in analysis, having the data stored and archived in their original state gives a common point of reference for derivative analyses. What constitutes sufficiently "raw" data is not always clear (e.g., ohms off a temperature sensor or images off an Illumina sequencing flowcell are generally not archived after the initial processing). Yet the spirit of this rule is that data should be as "pure" as possible when they are stored. If derivations occur, they should be documented by also archiving relevant code and intermediate datasets.

Rule 4: Store data in open formats

To maximize accessibility and long-term value, it is preferable to store data in formats whose specifications are freely available. The appropriate file type will depend on the data being stored (e.g. numeric measurements, text, images, video), but the key idea is that accessing data should not require proprietary software, hardware, or purchasing a commercial license. Proprietary formats change, maintaining organizations go out of business, and changes in license fees make access to data in proprietary formats unaffordable to end-users. Examples of open data formats include comma-separated values (CSV) for tabular data, hierarchical data format (HDF) and NetCDF for hierarchically structured scientific data, portable network graphics (PNG) for images, and extensible markup language (XML) for documents. Examples of closed formats include DWG (for AutoCAD drawings), Photoshop document (PSD, for bitmap images), Windows Media Audio (WMA, for audio recording files), and Microsoft Excel (for spreadsheets). Even if day-to-day processing uses closed formats (e.g., due to software requirements), data being stored for archival purposes should be stored in open formats. This is generally not prohibitive; most closed-sourced software enables users to export data to an open format.

Rule 5: Data should be uniquely identifiable

To aid reproducibility, the data used in a scientific publication should be uniquely identifiable. Ideally, datasets should have a unique identifier such as a Document

Object Identifier (DOI), Archival Resource Key (ARK), or a Persistant URL (PURL). An increasing number of online services, such as Figshare, Zenodo, or DataOne are able to provide these. Institutional initiatives also exist, and are known to your librarians.

Datasets evolve over time. In order to distinguish between different versions of the same data, each dataset should have a distinct name, which includes a version identifier. A simple way to do this is to use date stamps as part of the dataset name. Using the ISO 8601 standard avoids regional ambiguities: it mandates the date format `YYYY-MM-DD` (i.e. from largest time unit to smallest). For example, the date "February 1st, 2015", while written as `01-02-2015` in the UK and `02-01-2015` in the US, is the unambiguous `2015-02-01` under this standard.

Semantic versioning is a richer approach to solving the same problem [18]. The CellPack datasets are an example of this [19]. A semantic version number takes the form: `Major.Minor.Patch`, e.g. `0.2.7`. The *major version* numbers should be incremented (or bumped) when a dataset scheme has been updated, or some other change is made that is not compatible with previous versions of the data with the same major version number. This means that an experiment using version `1.0.0` of the dataset may not run on version `2.0.0` without changes to the data analysis. The *minor version* should be bumped when a change has been made which is compatible with older versions of the data with the same major version. This means that any analysis that can be performed on version `1.0.0` of the data is repeatable with version `1.1.0` of the data. For example, adding a new year in a temporal survey will result in a bump in the minor version. The *patch version* number is bumped when typos or bugs have been fixed. For example version `1.0.1` of a dataset may fix a typo in version `1.0.0`.

Rule 6: Link relevant metadata

It should be almost impossible to separate data from its associated metadata. The importance of metadata for context, reusability, and discovery has been written about at length in many guides for data best practices [9,13,20].

Metadata should be as comprehensive as possible, uses the relevant standards of your discipline, and be machine-readable (e.g., XML, JSON). Metadata should always accompany a dataset, wherever it is stored. How best to do this depends on the format of the data. Formats such as NetCDF or HDF5 allow for embedded metadata [21,22], so the data and metadata are bundled together. In a relational database, metadata tables should be clearly labeled and linked to the relevant data. Ideally a schema will be provided that also shows the linkages between data tables and metadata tables. Another scenario is a set of flat text files–in this case a semantically versionned, compressed archive should be created that includes metadata.

Whatever format is used for archiving, the goal should be to make the link between metadata and data as clear as possible. The best approach is dependent on the archiving plan used, but even if the dataset is archived solely for personal use, metadata will provide crucial context for future reuse.

Rule 7: Adopt the proper privacy protocols

In datasets where privacy is important, be sure to have a plan in place to protect data confidentiality. You should consider the different data stakeholders when developing privacy protocols for your data storage. These stakeholders include funding agencies, human subjects or entities, collaborators, and yourself. Both the NSF and NIH have

data sharing policies in their grant guidelines to prevent sharing personally identifiable information, and to anonymize data on human subjects.

In small datasets, a hashing scheme is enough to anonymize minimal personal information. Make sure to not store the hashing scheme with the data to prevent inadvertent sharing and don't use a commonplace hashing technique. Famously, New York City officials shared what they thought was anonymized data on cab drivers and over 173 million cab rides. However, it was quickly recognized that the city anonymized the data with a simple MD5 hashing scheme and all 20 GB of data was de-anonymized in a matter of hours [23]. In more problematic cases, the data itself allows identifiability: this is the case with human genomic data that map directly onto a subject's identity [24]. Methods for dealing with these complex issues at the intersection of data storage and privacy are rapidly evolving, and include storing changes against a reference genome to help with privacy and data volume [25,26], or bringing computation to data storage facilities instead of vice versa [27]. Having a plan for privacy before data acquisition is important, because it can determine or limit how data will be stored.

Rule 8: Have a systematic backup scheme

Every storage medium can fail, and every failure can result in loss of data. Researchers should therefore back data up at all stages of the research process. Data stored on local computers or institutional servers during the collection and analysis phases should be backed up to other locations and formats to protect against data loss. No backup system is failsafe (see the stories of the Dedoose crash and the near deletion of Toy Story 2), so more than one backup system should be used. Kristin Briney advocates the "Rule of 3" for backing up data: two onsite copies (such as on a computer, an external hard drive, or a tape) and one offsite copy (e.g. in cloud storage). For example, keeping backups in multiple locations protects against data loss due to theft or natural disasters.

Researchers should also test their backups regularly to ensure that they are functioning properly. Common reasons for backup failure include:

- faulty backup software
- incorrect configuration (e.g., not backing up sub-directories)
- encryption (e.g., someone has encrypted the backups but lost the password)
- media errors

Consider the backup plan of your selected data repository before publishing your data. Many repositories mirror the data they host on multiple machines. If possible, find out about the long-term storage plans of the repository. Are there plans in place to keep data available if the organization that manages the repository dissolves?

Rule 9: The location and method of data storage depends on how much you have.

The storage method you should choose depends on the size and nature of your data, the cost of storage, the time it takes to transfer the data, how the data will be used and any privacy concerns. Data is increasingly generated in the range of many terabytes by environmental sensors, satellites, automated analytical tools, simulation models, and genomic sequencers. Even larger data generating machines like the Large Hadron Collider (LHC) and the Large Scale Synoptic Survey Telescope (LSST)

generate many terabytes (TB) per day, rapidly accumulating to petabyte (PB) scale over the course of any particular study. While the cost of storage continues to decrease, the volume of data to be stored impacts the choice of storage methods and locations: for large datasets it is necessary to balance the cost of storage with the time of access and costs of re-generating the data.

When data takes too long to transfer or is costly to store, it can become more efficient to use a computer that can directly access and use the data in place. Inactive data can be put in longer-term storage; this is less expensive, but can take longer to retrieve. Some storage systems automatically migrate 'stale' files to longer term storage. Alternatively, some computing can be done 'in the database' or 'on disk' via database query languages (e.g. SQL, MapReduce) that perform basic arithmetic, or via the use of procedural languages (e.g. R, Python, C) embedded in the database server. Modern database technologies such as HDFS and Spark allow these computations to be done on data of almost any size. When data is larger than RAM, it can be handled by a 'big memory' node, which most high-performance computing have deployed – relying on tight software/hardware integration, these are currently around 1-4 TB. This allows the user to read in and use a large dataset without special tools.

If you regularly only need access to a small subset of your data or need to share it with many collaborators, a web based API (Application Programming Interface) might be a good solution. Using this method, many users can send requests to a web service which can subset the data, perform in-database computation, and return smaller volumes of data as specific slices. Tools based on web services make it easier to find and download data, and facilitate analysis via reproducible scripts, however they can lead to excessive and careless abuse of resources. The time required to re-download and recompute results can be reduced by 'caching'. Caching stores copies of downloads and generated files that are recognized when the same script is run multiple times.

Rule 10: Data should be stored in a machine readable-format

Not only data should be stored in an open format (Rule 4), but it should also be stored in a format that computers can easily use. This is especially crucial as datasets become larger. Machine readable data is best achieved by using standard data formats that have clear specifications (e.g., CSV, XML, JSON, HDF5), or by using databases. Such data formats can be handled by a variety of programming languages, as efficient and well-tested libraries for parsing them are typically available. These standard data formats also ensure interoperability, facilitate re-use, and reduce the chances of data loss or mistakes being introduced during conversion between formats.

When data can be easily imported into familiar software, whether it be a scripting language, a spreadsheet, or any other computer program that can import these common files, data become easier to re-use. Computer source code, the human readable software code that uses data, provide metadata as well. This makes the analysis more transparent, such that all assumptions are implicitly stated. This also enables extraction of the analyses performed, their reproduction, and their modification.

To take full advantage of data, it can be useful for it to be structured in a way that make manipulation and analysis easy. One such structure for data has been named *tidy* data [28]: each variable is a column, each observation is a row, and each type of observational unit is a table (Fig.˜1). When data is organized in this way, the duplication of information is reduced and it is easier to subset or summarize the dataset to include the variables or observations of interest.

A

species	habitat	weight	length	latitude/longitude	date
Alligator mississippiensis	swamp	431 lb	4 ft 2	29.531,-82.184	Sept 15, 2015
Puma concolor	forest	125 lb	2.2m	29.125,-81.682	08/10/2015
Ursus americanus	forest	88 kg	133 cm	N29°7'30"/W81°40'55.2"	07-13-2015

B

species_code	species_name
AM	Alligator mississippiensis
PC	Puma concolor
UA	Ursus americanus

station_code	habitat	latitude	longitude
1	swamp	29.531	-82.184
2	forest	29.125	-81.682

species_code	date	station	weight	length
AM	2015-09-15	1	196	127
PC	2015-08-10	2	57	220
UA	2015-07-13	2	88	133

Figure 1. Example of an untidy dataset (A) and its tidy equivalent (B). Dataset A is untidy because it mixes observational units (species, location of observations, measurements about individuals), the units are mixed and listed with the observations, more than one variable is listed in the coordinates for the observations (both latitude and longitude), several formats are used in the same column. Dataset B is an example of a tidy version of dataset A. Here, having species in a separate table is not necessarily needed but would allow researchers to add data, or change the name more easily than if the full species name was included in the measurement table. The tidy format also reduces the amount of information that is duplicated on each row.

Interoperability is facilitated when variable names are mapped to existing data standards. For instance, for biodiversity data, the Darwin Core Standard provides a set of terms that describe observations, specimens, samples, and related information for a taxa. Because each term is clearly defined and documented, each dataset can use the terms consistently, facilitating data sharing across institutions, applications, and disciplines.

With machine-readable data, it is also easier to build an Application Programming Interface (API) to query the dataset to retrieve a subset of interest as outlined in Rule 9.

Acknowledgements

National Center for Supercomputing Applications. Software Carpentry Foundation. iDigBio/NSF. May need some EPA language. Working on it..

Figure Legends

Figures here: Will need to figure out numbering...

Tables

Tables here: Will need to figure out numbering...

References

1. Reid JG, Carroll A, Veeraraghavan N, Dahdouli M, Sundquist A, English A, et al. Launching genomics into the cloud: deployment of Mercury, a next generation sequence analysis pipeline. BMC bioinformatics. 2014;15: 30. doi:10.1186/1471-2105-15-30

2. Hampton SE, Strasser C a, Tewksbury JJ, Gram WK, Budden AE, Batcheller AL, et al. Big data and the future of ecology. Frontiers in Ecology and the Environment. 2013; 130312142848005. doi:10.1890/120103

3. Eisenstein DJ, others. SDSS-III: Massive Spectroscopic Surveys of the Distant Universe, the Milky Way, and Extra-Solar Planetary Systems. The Astronomical Journal. 2011;142: 72.

4. Adams J. Collaborations: The rise of research networks. Nature. Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved. 2012;490: 335–6. doi:10.1038/490335a

5. Fraser LH, Henry HA, Carlyle CN, White SR, Beierkuhnlein C, Cahill JF, et al. Coordinated distributed experiments: an emerging tool for testing global hypotheses in ecology and environmental science. Frontiers in Ecology and the Environment. Ecological Society of America; 2013;11: 147–155. doi:10.1890/110279

6. Robertson MAG Tim AND Döring. The gBIF integrated publishing toolkit: Facilitating the efficient publishing of biodiversity data on the internet. PLoS ONE. Public Library of Science; 2014;9: e102623. doi:10.1371/journal.pone.0102623

7. Wolkovich EM, Regetz J, O'Connor MI. Advances in global change research require open science by individual researchers. Global Change Biology. 2012;18: 2102–2110. doi:10.1111/j.1365-2486.2012.02693.x

8. Roche DG, Lanfear R, Binning SA, Haff TM, Schwanz LE, Cain KE, et al. Troubleshooting public data archiving: suggestions to increase participation. PLoS biology. 2014;12: e1001779. doi:10.1371/journal.pbio.1001779

9. White E, Baldridge E, Brym Z, Locey K, McGlinn D, Supp S. Nine simple ways to make it easier to (re)use your data. Ideas in Ecology and Evolution. 2013;6: 1–10. doi:10.4033/iee.2013.6b.6.f

10. Goodman A, Pepe A, Blocker AW, Borgman CL, Cranmer K, Crosas M, et al. Ten simple rules for the care and feeding of scientific data. PLoS computational biology. 2014;10: e1003542. doi:10.1371/journal.pcbi.1003542

11. Pepe A, Goodman A, Muench A, Crosas M, Erdmann C. How Do Astronomers Share Data? Reliability and Persistence of Datasets Linked in AAS Publications and a Qualitative Study of Data Practices among US Astronomers. Golden AA-J, editor. PLoS ONE. 2014;9: e104798. doi:10.1371/journal.pone.0104798

12. Vines TH, Albert AYK, Andrew RL, Débarre F, Bock DG, Franklin MT, et al. The availability of research data declines rapidly with article age. Current biology : CB. Elsevier; 2014;24: 94–7. doi:10.1016/j.cub.2013.11.014

13. Michener WK, Jones MB. Ecoinformatics: supporting ecology as a data-intensive science. Trends in ecology & evolution. 2012;27: 85–93. doi:10.1016/j.tree.2011.11.016

14. Michener WK, Brunt JW, Helly JJ, Kirchner TB, Stafford SG. Nongeospatial metadata for the ecological sciences. Ecological Applications. Eco Soc America; 1997;7: 330–342.

15. Marcial LH, Hemminger BM. Scientific data repositories on the Web: An initial survey. Journal of the American Society for Information Science and Technology. 2010;61: 2029–2048. doi:10.1002/asi.21339

16. Wilson G. Software Carpentry: lessons learned. F1000Research. 2014;3: 62. doi:10.12688/f1000research.3-62.v1

17. Teal TK, Cranston Ka, Lapp H, White E, Wilson G, Ram K, et al. Data Carpentry: Workshops to Increase Data Literacy for Researchers. International Journal of Digital Curation. 2015;10: 135–143. doi:10.2218/ijdc.v10i1.351

18. Preston-Werner T. Semantic Versioning 2.0.0. http://semver.org; 2014.

19. Johnson GT, Goodsell DS, Autin L, Forli S, Sanner MF, Olson AJ. 3D molecular models of whole HIV-1 virions generated with cellPACK. Faraday Discuss. The Royal Society of Chemistry; 2014;169: 23–44. doi:10.1039/C4FD00017J

20. Strasser C, Cook R, Michener W, Budden A. Primer on Data Management : What you always wanted to know [Internet]. California Digital Libraries; 2012. doi:10.5060/D2251G48

21. Rew R, Davis G. NetCDF: An interface for scientific data access. Computer Graphics and Applications, IEEE. IEEE; 1990;10: 76–82.

22. Koziol Q, Matzke R. Hdf5–a new generation of hdf: Reference manual and user guide. National Center for Supercomputing Applications, Champaign, Illinois, USA, http://hdf ncsa uiuc edu/nra/HDF5. 1998;

23. Goodin D. Poorly anonymized logs reveal NYC cab drivers' detailed whereabouts [Internet]. 2015. Available: http://arstechnica.com/tech-policy/2014/06/poorly-anonymized-logs-reveal-nyc-ca

24. Homer N, Szelinger S, Redman M, Duggan D, Tembe W, Muehling J, et al. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS genetics. Public Library of Science; 2008;4: e1000167. doi:10.1371/journal.pgen.1000167

25. Kahn SD. On the future of genomic data. Science (New York, NY). 2011;331: 728–9. doi:10.1126/science.1197891

26. Wandelt S, Bux M, Leser U. Trends in genome compression. Current Bioinformatics. Bentham Science Publishers; 2014;9: 315–326.

27. Gaye A, Marcon Y, Isaeva J, LaFlamme P, Turner A, Jones EM, et al. DataSHIELD: taking the analysis to the data, not the data to the analysis. International journal of epidemiology. 2014;43: 1929–44. doi:10.1093/ije/dyu188

28. Wickham H. Tidy Data. Journal of Statistical Software. 2014;59: 1–23. Available: http://www.jstatsoft.org/v59/i10