Ten Simple Rules for Data Storage

Edmund Hart [1,*], Pauline Barmby [2], Jeffrey Hollister [3], David LeBauer [4],

1 Univeristy of Vermont, Department of Biology, Burlington
2 University of Western Ontario, Department of Physics and Astronomy
3 US Environmental Protection Agency, Atlantic Ecology Division
4 University of Illinois at Urbana-Champaign, National Center for Supercomputing Applications and Institute for Genomic Biology

∗ **E-mail: emh@emhart.info**

Abstract

Introduction

Some example text with a citation [1]

Rule 1: Rule

1 Rule 2: Rule {Know your use case}

Researchers should know their use case and store data appropriately. Is this data collected and just being archived? Will it change regularly? How will those changes be logged (e.g. provenance if any)? Will this be shared via an API? Linked to a paper? What are the institutional restrictions? Can you use a commercial service like Dropbox or use a personally maintained system? Knowing the reason why you're sharing your data will constrain your choices here.

Rule 3: Rule

Rule 4: Rule

Rule 5: Rule

Rule 6: Rule

Rule 7: Rule

Rule 8: Rule

2 Rule 9: Rule {Data size matters / requires special considerations}

- #39 and related GH issues #16, #19, #25

- Size classes:
  - larger than RAM
  - larger than HD space
  - larger than data storage server
- Storage method depends on the size of data; storage costs, transfer time, and computing costs can become substantial.
  - data generated by simulation and derived data should consider cost of storage vs. the cost of re-generating output.
  - For analyses of large data sets, the speed of reading and writing data can limt the speed of computation.
- Larger data sets that are actively used in analysis should be stored on a disk that is attached to a computer rather than being moved around between analysis and storage.
  - inactive data can be put in longer-term storage; this is less expensive, but can be slow to retrieve. Archiving of 'stale' files can be automated (and is at HPC centers).
- Data that is larger than memory can handle,
  - can be handled by 'big memory' nodes.
  - Computing can also be done 'in the database'
- Don't move (large data) around more than you have to - it can become inefficient, and make storage slower than necessary.
  - New tools make it easier to find and download data combined with reproducible scripts can lead to excessive and careless abuse of resources.
  - subset and compute on the server, in the database where possible. The dplyr R package does lazy eval; SQL can perform a wide range of data summaries, by groups, etc. On the other hand, it may be quicker to transfer normalized (e.g. 'flattening' a relational database can increase the size of data by orders of magnitude)
  - Use tools to store local 'cached' copies, instead of writing scripts that always download archived data. Only update data if there are changes. * knitr has a cache argument that saves time in re-computing and in re-downloading.
- For data larger than a single hard drive disk, up to multiple servers
  - requires a meta-data server to allow fast access to distributed across many disks
- For very large data
  - it is not practical to store data
  - there are trade offs among cost, information content, and accessibility.

Rule 10: Rule

3 Acknowledgements

National Center for Supercomputing Applications. Software Carpentry Foundation.

Figure Legends

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

Tables

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

References

1. 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. Public Library of Science; 2014;10: e1003542.