Chapter 1

DataDepsGenerators.jl: Making Reusing Data Easy by Automatically Generating DataDeps.jl Registration Code

This paper was originally published in the Journal of Open Source Software.

1.1 Summary

DataDepsGenerators.jl is a tool written to help users of the Julia programming language (Bezanson et al. 2014), to observe best practices when making use of published datasets. Using the metadata present in published datasets, it generates the code for the data dependency registration blocks required by DataDeps.jl (White et al. 2018). These registration blocks are effectively executable metadata, which can be resolved by DataDeps.jl to download the dataset. They include a message that is displayed to the user whenever the data set is automatically downloaded. This message should include provenance information on the dataset, so that downstream users know its original source and details on its processing.

DataDepsGenerators.jl attempts to use the metadata available for a dataset to capture and record:

- The dataset name.
- A URL for a website about the dataset.
- The names of the authors and maintainers.
- The creation date, publication date, and the date of the most recent modification.
- The license that the dataset is released under.
- The formatted bibliographic details of any paper about or relating to the dataset.
- The formatted bibliographic details of how to cite the dataset itself.
- A list of URLs where the files making up the dataset can be downloaded.
- A corresponding list of file hashes, such as MD5 or SHA256, to validate the files after download.

- A description of the dataset.

Depending on the APIs supported by the repository some of this information may not be available. DataDepsGenerators.jl makes a best-effort attempt to acquire as much provenance information as possible. Where multiple APIs are supported, it makes use of all APIs possible, merging their responses to fill any gaps. It thus often produces higher quality and more comprehensive dataset metadata than is available from any one source.

DataDepsGenerators.jl leverages many different APIs to support a very large number of repositories. By current estimates tens of millions of datasets are supported, from hundreds of repositories. The APIs supported include:

- DataCite / CrossRef
  - This is valid for the majority of all dataset with a DOI.
- DataOne
  - This supports a number of data repositories used in the earth sciences.
- FigShare
  - A popular general purpose data repository.
- Dryad
  - A data repository particularly popular with evolutionary biology and ecology.
- UCI ML repository
  - A data repository commonly used for small-medium machine learning benchmark datasets.
- GitHub
  - Most well known for hosting code; but is fairly regularly used to host versioned datasets.
- CKAN
  - This is the system behind a large number of government open data initiatives such as Data.Gov, data.gov.au, and the European Data Portal.
- Embedded JSON-LD fragments in HTML pages.
  - This is commonly used on many websites to describe their datasets. Including some of those listed above; as well as Zenodo, Kaggle Datasets, all DataVerse sites and many others.

DataDepsGenerators.jl as the name suggests, generates static code which the user can add into their project's Julia source code to make use of with DataDeps.jl. There are a number of reasons why static code generation is preferred over directly using the APIs.

- On occasion the information reported by the APIs is wrong, incomplete or overly detailed. The user may tweak the details as required by editing the generated code.
- The process of accessing the APIs requires a number of heavy dependencies, such as HTML and JSON parsers. If these APIs were to be accessed directly by a project, it would require adding this large dependency tree to the project.
- It is important to know if a dataset has changed. As such retrieving the file hash and last modification date would be pointless if they are updated automatically.

Finally: having the provenance information recorded in plain text, makes the dataset metadata readily accessible to anyone reading the source code; without having to run the project's application.

The automatic downloading of data is important to allow for robustly replicable scientific code. The inclusion of provenance information is required to give proper credit

and to allow for good understanding of the dataset's real world context. DataDepsGenerators.jl makes this easy by automating most of the work.

1.1.1 Other similar packages

In the R software ecosystem there is the `suppdata` (Pearse and Chamberlain 2018) package. `suppdata` is a package for easily downloading supplementary data files attached to journal articles. It is thus very similar in purpose: to make research data more accessible. It is a direct download tool, rather than DataDepsGenerators.jl's approach of generating metadata that is evaluated to preform the download. While there is some overlap, in that both support FigShare and Dryad, `suppdata` primarily supports journals rather than data repositories.

When it comes to accessing data repositories, there exists several R packages which only support a single provider of data. These vary in their support for different functionality. They often support features beyond the scope of DataDepsGenerators.jl, to search, or upload data to the supported repository. Examples include:

- `rdryad` for DataDryad
- `rfigshare` for FigShare
- `ckanr` for CKAN
- `rdatacite` for DataCite
- `rdataone` for DataOne

To the best of our knowledge at present there does not exist a unifying R package that supports anywhere near the range of data repositories supported by DataDepsGenerators.jl. Contemporaneously, with the creation of DataDepsGenerator.jl, there was a proposal for another related R package (`doidata`) which would access data based on a DOI. While this has yet to eventuate into usable software, several of the discussions relating to it were insightful, and contributed to the functionality of DataDepsGenerators.jl.

1.1.2 Acknowledgments

This work was largely carried out as a Google Summer of Code project, as part of the NumFocus organisation. It also benefited from funding from Australian Research Council Grants DP150102405 and LP110100050.

We also wish to thank the support teams behind the APIs and repositories listed above. In the course of creating this tool we thoroughly exercised a number of APIs. In doing so we encountered a number of bugs and issues; almost all of which have now been fixed, by the attentive support and operation staff of the providers.