IJDC | *General Article*

Frictionless Data: Making Research Data Quality Visible

Dan Fowler
Open Knowledge International

Jo Barratt
Open Knowledge International

Paul Walsh
Open Knowledge International

Abstract

There is significant friction in the acquisition, sharing, and reuse of research data. It is estimated that eighty percent of data analysis is invested in the cleaning and mapping of data (Dasu and Johnson, 2003). This friction hampers researchers not well versed in data preparation techniques from reusing an ever-increasing amount of data available within research data repositories. Frictionless Data is an ongoing project at Open Knowledge International focused on removing this friction. We are doing this by developing a set of tools, specifications, and best practices for describing, publishing, and validating data. The heart of this project is the "Data Package", a containerization format for data based on existing practices for publishing open source software. This paper will report on current progress toward that goal.

Received 20 October 2016 ~ *Accepted* 23 January 2017

Correspondence should be addressed to Jo Barratt, 3rd Floor, 86-90 Paul Street, London, EC2A 4NE, United Kingdom. Email: jo.barratt@okfn.org

An earlier version of this paper was presented at the 12th International Digital Curation Conference.

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by the University of Edinburgh on behalf of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

Copyright rests with the authors. This work is released under a Creative Commons Attribution Licence, version 4.0. For details please see https://creativecommons.org/licenses/by/4.0//

http://dx.doi.org/10.2218/ijdc.v12i2.577
DOI: 10.2218/ijdc.v12i2.577

Introduction

Sharing data for reuse by other researchers is an increasing norm within scientific practice. It helps build a sense of trust in published results, promotes transparency, and encourages reproducibility (Peng, Dominici and Zeger, 2006). Open Knowledge International has long advocated for "open" approaches to civic engagement through the development of tools, standards, and best practices for sharing data. We have promoted awareness of these practices in scientific research through forums such as the Working Group on Open Data in Science (Molloy, 2011). Despite growing awareness of the benefits of sharing research data, there are often legal, social, and technical barriers. Through our Frictionless Data project, we have investigated the extent to which we can address some of the technical barriers through a non-proprietary, interoperable, extensible, and distributed approach to sharing data called the Data Package.

The Data Package is a format for storing useful metadata alongside a dataset expressed as a simple JSON file named datapackage.json. The Data Package, with respect to research, is not about long-term preservation, citation, or documentation of research processes or protocols. Rather, it focuses specifically on the import and publishing stages of a data analysis project.[1]

Why Does Open, Granular Metadata Matter?

We currently focus on packaging data that naturally exists in "tables" – for example, CSV files – a clear area for enrichment illustrated by guidelines issued by Wellcome Open Research. The guidelines mandate:

> 'Spreadsheets should be submitted in CSV or TAB format; EXCEPT if the spreadsheet contains variable labels, code labels, or defined missing values, as these should be submitted in SAV, SAS or POR format, with the variable defined in English' (Wellcome Open Research, 2016).

Guidelines like these typically mandate that researchers submit data in non-proprietary formats; SPSS, SAS, and other proprietary data formats are accepted due to the fact they provide important contextual metadata that haven't been supported by a standard, non-proprietary format. The Data Package specifications – in particular, our Table Schema specification – provide a method of assigning functional "schemas" for tabular data which define expected types, enumerated values, constraints (e.g. maximum and minimum expected values for columns), and relations between columns in an attempt to address this.

One reason researchers might be disincentivized to share data is the fear that it will be "misinterpreted" due to the complexity of the data (Tenopir et al., 2011). In the absence of good metadata, misinterpretation of even simple datasets is not only possible, but likely given the heuristic nature of automatic type inference in data analysis programs.

In one example of such an issue, Zeeberg et al. (2004) and later Ziemann, Eren and El-Osta (2016) describe a phenomenon where gene expression data was silently corrupted by Microsoft Excel:

[1] The full specifications can be found at http://specs.frictionlessdata.io

> ‘A default date conversion feature in Excel (Microsoft Corp., Redmond, WA) was altering gene names that it considered to look like dates. For example, the tumor suppressor DEC1 [Deleted in Esophageal Cancer 1] [3] was being converted to “1-DEC”’ (Zeeberg et al., 2004).

These errors didn’t stop at the initial publication, but rather propagated through multiple data repositories. Zeeberg et al. (2004) describe various technical workarounds to avoid this problem, including using Excel’s text import wizard to manually set column types every time the file is opened. A simple, open, and ubiquitous method to unambiguously declare types in columnar data – columns containing gene names (e.g. “DEC1”) are strings not dates and “RIKEN identifiers” (e.g. “2310009E13”) are strings not floating point numbers – paired with an Excel plugin that reads this information may be what is needed. By providing this granular metadata with the data, both users and software programs can use it to automatically import into Excel, R, SQL, or other data analysis software without having to resort to manual, error-prone processes.

Similar Initiatives

There are several other data “packaging” initiatives that currently exist that share a “files on disk” model for pairing data with metadata. These include BagIt (Kunze et al., 2016) and W3C’s CSV on the Web (CSVW) (W3C, 2016). BagIt is a dataset packaging format popular among libraries, including the Library of Congress in the US, designed to support disk-based storage and network transfer of arbitrary digital content. BagIt focuses on archival of digital assets (primarily content) while Data Package focuses on describing the structure of packaged data. In addition, a key element of the Data Package approach is integration with existing tooling and extensibility to many types of data, both of which are not priorities for BagIt.

CSVW, the result of a W3C working group concluded in 2016, is more directly comparable. CSVW started as an effort to standardize the then-prototype Tabular Data Package specification with several modifications to ensure web-friendliness. Given the degree of overlap in our approaches and supporting tools, we imagine increasing crossover in tool and specification support as the Data Package specifications and related tooling mature.

Progress

Work on the Data Package specifications and related tooling has been in progress for several years. In 2016, supported by the Alfred P. Sloan Foundation, we have worked to reach version 1.0 of the specifications while creating a set of tools informed by real experience. To do this, we have identified and worked with researchers in fields such as as computational biology, energy modeling, and neuroscience who have expressed interest in exploring proof-of-concept pilots. We have also documented the experiences of groups and individuals who had already adopted the specifications through a series of case studies. In addition, we have initiated several technical partnerships with organizations such as the Open Data Institute and rOpenSci (Ram, 2013) to develop software libraries and tools. Understanding the need for broad awareness to encourage adoption, we have promoted this approach through presentations, blog posts, and tutorials at research events. At the same time, we have continued developing our core Python and JavaScript tooling.

Case Studies

To help capture some of the motivations, successes, and challenges of those already using the specifications, we have started a case study series[2]. We interviewed the developer of Dataship, a social platform for data analysis; a developer at Tesera, a software consultancy; and researchers at Open Power System Data, a collective researching methods of publishing high quality and legally open energy data online. Two main themes emerged: (1) Data Packages provide software developers with a standard container for passing data in cloud-centric workflows and (2) Data Packages provide a useful method to self-publish research data with clear metadata.

Standard container for cloud-based applications

The developer of Dataship, a social network for analyzing data via Jupyter-style notebooks, adopted the Data Package as the base format for these notebooks rather than build his own ad hoc data structure. This was, in part, to build on a growing set of tools such as dpm, a command-line tool for downloading Data Packages:

> ‘Every notebook on Dataship is also a Data Package. Like other Data Packages it can be downloaded, along with its data, just by giving its URL to a tool like dpm’ (Waylon Flinn, Dataship).

Similarly, developers at Tesera used the specifications, paired with CSV files, as an intermediary format for passing data from a variety of proprietary sensors – including those for Light Detection and Ranging (LiDAR) and color infrared – around various services such as those provided by Amazon.

> ‘This helps us ensure good interoperable data at a foundational level, thereby making it easier to use for analysis, visualization, or modeling without extensive ad hoc quality control’ (Spencer Cox, Tesera).

These projects highlight the utility and flexibility of the Data Package formats in a variety of developer-friendly, research-adjacent use cases. They are also provide examples of the growing ecosystem of externally developed tools and platforms that support the specifications.

High quality publishing format for research data

Open Power System Data’s use of the specifications demonstrates a use case for working researchers. Open Power System Data[3] is a free-of-charge and open platform providing clean, high quality data needed for power system analysis and modeling (Lion Hirth and Ingmar Schlecht, Open Power System Data). The platform provides a way to harmonize poorly structured energy datasets across a variety of formats, file types, and indicators for exposing missing or invalid data. In addition, licensing issues are a core concern for this project; the specifications provides a clear, machine-readable way for data publishers to assign a license to their data.

Learnings

Overall, these case studies point to an early positive reaction from groups with direct or indirect research use cases. What remains to be explored is how Data Packages

[2] Frictionless Data Case studies: http://frictionlessdata.io/case-studies/

[3] Open Power System Data: http://open-power-system-data.org

provided through these platforms are used. This is partly dependent on further development of software integrations, improvements to our existing tools, greater awareness, and wider deployment of Data Packages across research data repositories.

Pilots

We have pursued proof-of-concept pilots to explore various research data use cases in fields like energy data, computational biology, and archaeology. Having started at the end of 2016, most of these pilots are only just underway. We look forward to sharing more outputs of each pilot as time goes on.

Data management for TEDDINET

As part of the Data Management for TEDDINET (DM4T), Open Knowledge International are working with researchers at the University of Bath to pilot the use of Frictionless Data specifications and tooling for datasets created through TEDDINET (Transforming Energy Demand through Digital Innovation NETwork)[4]. The goal of the pilot is to demonstrate this approach to preparing and publishing research data to facilitate greater re-use. This pilot is being conducted in the open on GitHub[5].

In an initial example, a dataset containing Electrical Load Measurements was “packaged” using our Python library: the dataset’s README, which contained a data dictionary, was encoded into a datapackage.json without altering the CSVs that comprise the dataset. This allows the full, multi-CSV dataset to be loaded into R at one time, using our Data Package library for R for analysis.

Active Data Biology at Pacific Northwest National Laboratory

We are working with a computational biologist at Pacific Northwest National Laboratory (PNNL) who is developing a collaborative data analysis tool for biological data called Active Data Biology. Data and analysis scripts for studies in the project are stored in GitHub. In the repository, a datapackage.json file has been created for the core metadata.tsv file. This describes the format of the file as well as the types and value constraints of the dataset. Through the Data Package, we have defined “missing values” for columns and types. As an example, here is the field metadata for the primary_therapy_outcome_success column of metadata.tsv:

```
{
  "constraints": {
    "enum": [
      "COMPLETE RESPONSE",
      "PARTIAL RESPONSE",
      "PROGRESSIVE DISEASE",
      "STABLE DISEASE"
] },
  "description": "it means the primary treatment a
success
                  (the patient survived)",
  "missingValues": [
    "[Not Applicable]",
    "[Not Available]",
```

[4] TEDDINET: https://teddinet.org/

[5] DM4T Pilot GitHub Repository: https://github.com/frictionlessdata/pilot-dm4t

```
        "[Pending]"
      ],
      "name": "primary_therapy_outcome_success",
      "type": "string"
    }
```

This pilot is directly informing work on a data validation service called “GoodTables.io” (see below) to automatically validate and flag errors on entry, maintaining data integrity over the life of a study and beyond. This pilot is being conducted in the open on GitHub[6].

The Western Pennsylvania Regional Data Center

The Western Pennsylvania Regional Data Center (WPRDC) provides a shared technological and legal infrastructure to support research, analysis, decision making, and community engagement. We will be working with the University of Pittsburgh Center for Urban and Social Research to pilot a number of tools on the portal, including the Data Quality Dashboard, Good Tables and integrations with CKAN.

Archaeology Working Group

In addition to our more formal pilots, drawing on the strength of the Open Knowledge Network, we have started to work the Open Archaeology Working Group to explore open approaches to working with archaeological data. In the latter part of 2016, we have worked with the group to pilot an approach to storing and manipulating archaeological data in Data Packages.

Technical Work

Specifications working group

The working group has met monthly ensure the specifications are as useful as possible to a wide range of users of users by reaching version 1.0 by the end of 2016. The working group is curated by Rufus Pollock as well as the Frictionless Data Technical Team at Open Knowledge and includes representatives from DAT project, Tesera Systems, Link Digital, Open North and the University of Washington. Version 1.0 of the specifications was released on 23rd January 2017, a significant milestone for the Frictionless Data team, and for the community of consumers of our specifications and tooling. Notable aspects of the v1 specification release include the following:

- The removal of many examples of unclear/ambiguous wording making it hard to implement specifications in certain corner cases;
- The delivery of around 20 core fixes and enhancements targeted for version 1.0;
- A redesign of how we write the specifications to make it easy to generate RFC versions for the IETF;
- The addition of a new specification for “Data Resource”, which was previously only specified as part of “Data Package”. The distinct Data Resource specification addresses an extremely common use case / pattern that we have seen in our own work and that of others;

[6] PNNL GitHub Repository: https://github.com/frictionlessdata/pilot-pnnl

- The renaming of JSON Table Schema to Table Schema, and the simplification of some important aspects of that specification;
- The official registration of media types for the specifications with IANA, which was confirmed on 11th January 2017. This is the first step in official registration of the specifications with global bodies. Next, we are submitting all specifications as RFCs with the IETF. We have registered the following media types:
  - application/vnd.datapackage+json
  - application/vnd.dataresource+json
  - application/vnd.tableschema+json

Setting a deliverable of version 1.0 for the beginning of 2017 forced us to look critically on the ambiguity that had built up in the specifications over the past several years, and allowed us to clarify elements based on real world usage in our own work and in the initial piloting period. The specifications are much stronger for this process, and we have an excellent foundation for swift iteration over 2017. The next steps from this work are to update our core implementations to support the new changes, and to identify a minimal set of enhancements to discuss towards a version 1.1 release in three months.

Internal implementations

We are working on Python and JavaScript libraries that implement the specifications and meet the needs of our pilot partners and the wider community. The next steps for our internal implementations is to update them to fully support version 1.0 of the specifications. Afterwards, there will continue to be maintenance and bug fixing, but the majority of our development efforts will shift “higher up the stack” to the applications we can build for users on top of this base, and as validated via our piloting.

- Data Package (Python)[7]
- Data Package (JavaScript)[8]
- Data Package JSON Table Schema (Python)[9]
- Data Package JSON Table Schema (JavaScript)[10]

Additional Libraries

- Good Tables (Python)[11]
- CKAN Extension for importing and exporting Data Packages[12]
- Generate SQL tables, load and extract data, based on JSON Table Schema descriptors[13]

[7] Data Package (Python): https://github.com/frictionlessdata/datapackage-py

[8] Data Package (JavaScript): https://github.com/frictionlessdata/datapackage-js

[9] Data Package JSON Table Schema (Python): https://github.com/frictionlessdata/jsontableschema-py

[10] Data Package JSON Table Schema (JavaScript): https://github.com/frictionlessdata/jsontableschema-js

[11] Good Tables (Python): https://github.com/frictionlessdata/goodtables-py

[12] CKAN Extension: https://github.com/ckan/ckanext-datapackager

[13] JSON Table Schema – SQL (Python): https://github.com/frictionlessdata/jsontableschema-sql-py

- Generate BigQuery tables, load and extract data, based on JSON Table Schema descriptors.[14]

Partner implementations

In addition to the work carried out by the Open Knowledge International technical team, we have worked with a number of technical partners on developing additional libraries to meet the needs of our pilot partners and the wider community.

rOpenSci are the official maintainers of the R libraries for (Tabular) Data Packages and will be the steward of these R libraries. This work has been completed and we have an agreement to build on and maintain these libraries up to the end of 2017.

- Data Package R Library[15]

Open Data Institute (ODI) Labs are the official maintainers of the Ruby libraries. The initial work has been completed and they will continue to provide maintenance and support to the libraries. Moving forward we are looking to collaborate with the ODI Labs on a number of tools for sharing and working with data.

- Data Package Ruby Library[16]
- Data Package JSON Table Schema Ruby Library[17]

Third-party implementations

An encouraging sign of the usefulness of the specifications is the adoption of them without our explicit mediation. Some examples include MetaTab for spreadsheets (MetaTab, 2017), and Laravel Datasets[18] which integrates some of our work into a popular web application framework written in PHP. SmartCSV.fx is an application for editing CSV files first developed around November of 2015. The purpose, according to its creator, was to ease the creation of high quality CSV files:

> ‘At work I have the need to fix wrong CSV files from customers. It is hard to find the errors and fix them in a text editor, even in a “normal” CSV editor. So I decided to write this simple JavaFX application’ (Billman, 2017b).

The tool originally used a custom CSV schema definition devised by the creator. Given its overlap of purpose, we reached out to the developer to better align his work with ours. Within a short amount of time, the developer switched to using Table Schema (Billman, 2017a). This development is on track to provide a Java-based Table Schema library and, possibly, a Data Package library in 2017.

[14] JSON Table Schema - BigQuery (Python): https://github.com/frictionlessdata/jsontableschema-bigquery-py

[15] Data Package R Library: https://github.com/frictionlessdata/datapackage-r

[16] Data Package Ruby Library: https://github.com/frictionlessdata/datapackage-rb

[17] Data Package JSON Table Schema Ruby Library: https://github.com/frictionlessdata/jsontableschema-rb

[18] Laravel Datasets: https://github.com/bluora/laravel-datasets-okfn

Driving data quality through user-facing apps

On top of the core implementations, there are three major applications we are building leveraging our specifications and libraries: GoodTables.io, the Data Quality Dashboard, and the Data Package Registry. All three aim to solve painful, real-world problems when working with tabular data from a variety of sources. All three applications are concerned with the validation and promotion of data quality, and the enhanced possibilities for data usage based on a standardized, quality assured base.

GoodTables.io is a web service for the continuous validation of data quality. Modeled on the popular and effective "continuous integration" paradigm from software engineering, GoodTables.io allows any user working with data to register a data set, and have it automatically checked for quality on every update. GoodTables.io is based directly on our work in goodtables-py, which is the library that provides the majority of the logic. It is already usable as alpha software, and is part of our pilot with the Pacific Northwest National Laboratory (see above). A public beta should be ready for use in early 2017, following which we will add more data repository integrations (currently, data must be published on either GitHub or Amazon S3).

The Data Quality Dashboard is a service to interact with and visualize a set of data quality results. The core functionality is built on top of Good Tables, Data Package, and Table Schema, and it can work with any collection of data sources, tracking progress and generating high level statistics over time. Our pilot partner, The Western Pennsylvania Regional Data Center, is employing the dashboard over a large set of research data. The Dashboard currently also features a dedicated integration with CKAN and this has led to adoption by a number of existing CKAN portals in government and elsewhere. The next steps are to update the service so it can run in a self-service, automated fashion.

Together with our technical partner, Atomatic, we have developed the Data Package Registry: a web app for the storage and retrieval of Data Packages. It provides a simple pathway for publishing data, with built-in, high-level views via simple tables and charts, using the forthcoming Data Package Views specification. The work as part of this grant is building on the original Data Package Registry. While there is nothing about Data Packages that requires a centralized registry for their publication, this centralized registry does seek to become a high quality resource for open data across a range of fields of interest.

We are working with the UK Data Service, to explore the use of the registry as a way to replace current publication flows. The work on the first release of the registry is expected to finish in February 2017. It brings together all our specification and core implementation work, and exposes a smooth publication flow for data publishers, from source data right through to visualization of that data.

Conclusion

Having started work on this concept through the process of developing CKAN, OpenSpending, and other data-intensive civic technology projects, we believe a decentralized, open standard for publishing tabular research data building on existing formats like CSV and JSON is a substantial contribution. Our experiences so far point to an unmet need for exactly this kind of approach in the research data ecosystem. Over the last year, we have noticed a very positive reaction driven by the needs of toolmakers (e.g. keeping the standards as simple as possible to make them easy to implement) while learning as much as we can about the needs of working researchers.

It is sometimes said that data standards are like toothbrushes: a good idea but no one wants to use anyone else's. Many existing standards and best practices already exist for sharing research data, however, as Ball et al. (2014) note:

> 'the greatest problems occur, though, where standards compete directly and in the case of metadata standards this is rarer than might first be apparent.'

Our work with various communities, researchers, developers, and others, it is clear that the Data Package specifications are useful, not just for researchers, but for anyone who works with tabular data.

Through this approach, we expect broad-based improvements in data quality as well as increased re-use of data. As expressed by a researcher in our case study with Open Power System Data, significant time and energy is currently lost to cleaning data by early career researchers, many of whom may be more interested in generating novel insights than the sometimes tedious mechanics of data "wrangling". By providing an enabling environment for tools to create and consume well-packaged data, we can empower these researchers to do more with less by allowing for the integration of modular, automated data import and validation services into research data repositories. We suggest that data quality can thereby be made "visible" by enabling better quality control and providing standardized visualization options through tools like the forthcoming GoodTables.io, Data Quality Dashboard, and Data Package Registry.

Going Forward

We will continue to support users of the specifications while seeking out new research-specific pilots and case studies. This will further allow us to develop the specifications and tooling grounded in lived reality. For example, Data Retriever, a tool developed by the "Weecology" lab[19] recently adopted the Data Package specification. The tool automates the finding and restructuring of ecological datasets. White et al. (2013) has argued "much of the shared data in ecology and evolutionary biology are not easily reused because they do not follow best practices in terms of data structure, metadata, and licensing." We will seek to interview the team to elaborate on their motivations for adoption and any challenges they faced. We have also seen interest by members of the Digital Humanities community. For instance, the Carnegie Museum of Art made their collections records public as a Data Package[20].

By providing a simple metadata format that also describes tabular data at a columnar level, we hope to enable data transport integrations across a diversity of platforms, from the cloud (Google's BigQuery and Amazon's AWS) all the way to the researcher's desktop R, Python Pandas, or SQL environment. In order to get there, we need better support and integrations with the tools researchers use; Data Package integrations in software packages like MATLAB, SPSS, and SAS are critical. While early community-built plugins exist, we need more researchers to provide solid use cases for piloting, further development, iteration, and testing. This also applies to the development of our specifications. Most of the relevant specifications and tooling work on Frictionless Data can be found on our GitHub organization[21]; we welcome contributions.

[19] Weecology Lab: http://weecology.org/

[20] See https://github.com/cmoa/collection

[21] Frictionless Data – GitHub: https://github.com/frictionlessdata/

Acknowledgments

The development of Frictionless Data was generously supported by the Alfred P. Sloan Foundation.

References

Ball, A., Chen, S., Greenberg, J., Perez, C., Jeffery, K., & Koskela, R. (2014). Building a disciplinary metadata standards directory. *International Journal of Digital Curation 9*(1). doi:10.2218/ijdc.v9i1.308

Billmann, A. (2017a) SmartCSV.fx version 0.9. Retrieved from http://www.billmann.de/2016/10/19/smartcsv-fx-version-0-9/

Billmann, A. (2017b). SmartCSV.fx. Retrieved from https://github.com/frosch95/SmartCSV.fx

Dasu, T., & Johnson, T. (2003). Exploratory data mining and data cleaning. Vol. 479. John Wiley & Sons.

Kunze, J., Littman, J., Madden, L., Summers, E., Boyko, A., Vargas, B. (2016). The BagIt file packaging format (V0. 97). Retrieved from https://tools.ietf.org/pdf/draft-kunze-bagit-12.pdf

Metatab. (2017). Metatab interface to datapackage.json. Retrieved from http://metatab.org/2016/10/25/metatab-interface-to-datapackage-json/

Molloy, J.C. (2011) The Open Knowledge Foundation: Open data means better science. *PLOS Biology, 9*(12). doi:10.1371/journal.pbio.1001195

Peng, R.D., Dominici, F., & Zeger, S.L. (2006). Reproducible epidemiologic research. *American Journal of Epidemiology, 163*(9).

Ram, K. (2013). *rOpenSci - open tools for open science.* AGU Fall Meeting Abstracts.

Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A.U., Wu, L., Read, E., Manoff, M., & Frame, M. (2011). Data sharing by scientists: Practices and perceptions. *PLOS ONE 6*(6). doi:10.1371/journal.pone.0021101

W3C CSV on the Web Working Group. (2016). CSV on the Web Repository. Retrieved from http://w3c.github.io/csvw/

Wellcome Open Research. (2016). How to publish - data guidelines. Retrieved from https://wellcomeopenresearch.org/for-authors/data-guidelines

White, E.P., Baldridge, E., Brym, Z.T., Locey, K.J., McGlinn, D.J., & Supp, S.R. (2013). Nine simple ways to make it easier to (re) use your data. *Ideas in Ecology and Evolution 6*(2).

Zeeberg, B.R., Riss, J., Kane, D.W., Bussey, K.J., Uchio, E., Lineham, W.M, Barrett, J.C., & Weinstein, J.N. (2004). Mistaken identifiers: Gene name errors can be introduced inadvertently when using Excel in bioinformatics. *BMC Bioinformatics 5*(1).

Ziemann, M., Eren, Y., & El-Osta, A. (2016). Gene name errors are widespread in the scientific literature. *Genome Biology, 17*(1). doi:10.1186/s13059-016-1044-7