PharmaSUG 2021 - Paper AD-176

Open-Source Development for Traditional Clinical Reporting

Mike Stackhouse, Nathan Kosiba, Atorus Research

ABSTRACT

As the pharmaceutical industry continues to embrace open-source solutions, gaps between the open-source ecosystem and the requirements of traditional clinical reporting pipelines continue to present themselves. These gaps are understandable, as regulatory requirements and decades of tradition cannot match up to the speed and freedom of software designed by consensus and open for review. Furthermore, with open-source as a novel concept within clinical reporting, contributors and consumers within this arena are sparse. Therefore, open-source solutions are not currently tailored to the specific needs of the clinical reporting pipeline. The way to close these gaps is not to wait or rely on the open-source contributors outside industry to close them for us, but rather to close them ourselves. This presentation will discuss the ongoing open-source work at Atorus, and how collective open-source development within the pharmaceutical industry can benefit both individual organizations and the industry as a whole.

INTRODUCTION

What if you could download an analysis reporting pipeline from GitHub? It could be some source other than GitHub – maybe The Comprehensive R Archive Network (CRAN), PyPi, or something totally new. Regardless of the source, what if companies did not have to establish and build these processes on their own, but could rely on publicly available resources to establish their tooling for an analysis data preparation and reporting pipeline?

Before going any further, it is important to understand why open-source tools can be beneficial in the first place. Within industry, there are many common challenges we face across organizations. Compare the industry today to the industry prior to introduction of CDISC. Before CDISC, data structures were developed on a company-by-company basis. Each company had to develop their own standard, staff had to be trained, and the agencies had to interpret those standards during a submission. This was redundant, as each organization was largely facing the same challenges yet developing different solutions.

CDISC was able to create a unified approach, and the benefits have been quite tangible. Instead of organizations developing their own standards, there was guidance to follow. Staff could learn one standard, and those skills then became transferrable, lowering onboarding times. Most importantly, it allows the agency to streamline their review process (Chen 2020). With a unified approach, the work was done and no longer had to be repeated within individual organizations.

Within our programming processes, industry still faces a similar challenge. A lot of code produced is redundant. Companies develop reporting pipelines from the ground up to ultimately create very similar outputs, both in data and reports. If the code is so redundant, then why reinvent the wheel?

POWERFUL OPEN-SOURCE TOOLS

The open-source ecosystem has a great deal to offer the clinical industry. The statistical programming language R has been at the forefront of open-source adoption over the past few years. Examining the language makes it easy to see why it is so appealing. As a free and open-source language, R has a wide array of statistical capabilities out of the box. With a massive repository of third party contributed packages within networks like CRAN or Bioconductor, there is a very strong chance that implementations of cutting-edge statistical methods are already available for download.

Another key factor in the adoption of R is the availability of tool sets like the tidyverse. A common criticism of open-source programming languages is that with the wide variety of third-party contributors, combinations of separate packages tend to lack a holistic design paradigm. The syntax and use of one package may vary greatly from another. In this respect, commercial languages benefit from greater

control over available tools within the language. Within R, the tidyverse offers an extended set of tools that follow an opinionated design. All packages share an underlying design philosophy, grammar, and data structures (Tidyverse, 2019). This is significant in that these are powerful tools that significantly expand tools within base R. This well supported ecosystem of packages expands further into other sets such as r-lib, tidymodels, R Markdown, and Shiny.

With all of these tools available, why has the industry yet to widely adapt R? The tidyverse offers powerful data manipulation with the packages dplyr and tidyr. R is extensively used in research and offers multiple packages to make publication ready tables, such as huxtable and gt. R even has powerful graphics tools to produce static, and even interactive graphs – and host them on the web using Shiny. R Markdown allows you to create reproducible reports in PDF, HTML, and more.

It is important to remember that the process of preparing data and running an analysis is only a small piece of the puzzle. Changing the requirements or the products midway through a pipeline can cause upstream or downstream disruptions that can cause more problems than value. This is not to say that the advanced tools available with R are of no value; there is immense potential of what these tools can offer the industry as modern data science are continually being adopted. Rather, if industry is to adopt these tools then the focus must shift to fitting them to the processes we follow today.

OPEN-SOURCE DEVELOPMENT FOR TRADITIONAL CLINICAL REPORTING

This is an area where R has a weakness that is continually becoming less and less significant. The clinical industry has several common practices that are quite industry specific. For example, the use of RTF file formats is prevalent, when much richer and more flexible file formats such as DOCX or HTML are quite well supported. Furthermore, the use of SAS® Version 5 transport files are still a required part of an FDA submission, despite a wide variety of more modern and less restrictive file formats.

It can be difficult to grasp why some of these requirements exist without looking at the bigger picture. Statistics and Reporting sits in the middle of a larger pipeline. This starts at source data collection and moves through data management before data are received by Statistics and Reporting. The outputs of this group are then handed to medical writing and delivered to regulatory.

This begins to explain why RTF formats are still common practice. The benefit of RTF is that it is easily opened within Microsoft Word, which is the common tool of choice for medical writing. RTF is a well-supported output format in SAS®, and thus has been the common output format of choice for many organizations. Many organizations also build out larger pipelines and tools on top of these processes. For example, organizations may have tools that combine all the necessary RTF files for tables, listing, and figures into PDF format for entry into a CSR. Changing the file format for a delivery can disrupt these processes downstream.

OPEN-SOURCE PACKAGE DEVELOPMENT AT ATORUS

RTF output is something that has understandably been less supported in R than modern formats. In the area of table production, the tidyverse package gt has recently been making great strides towards implementing RTF support. Outside of gt, the package huxtable allows for most necessary table formatting options in RTF as well – but both packages lack some critical features for the clinical reporting pipeline, such as:

- The ability to change document orientation
- Repeating titles and column headers from page to page
- Footnotes attached to the bottom of the page

For this reason, Atorus developed the open-source R package pharmaRTF. pharmaRTF solves these specific issues by allowing a user to format a table using the packages that support RTF already but supplement the gaps that are missing. Users can set document properties such as orientation, page size, and margins. They can provide the multi-line titles and footnotes that are common within clinical tables, including special formatting like page numbers, or split headers where text is aligned to both sides.

Furthermore, column headers from the table are repeated across pages. This effectively allows a user to reproduce statistical tables in the style and output format required for clinical tables using R.

Once capabilities within R have been established, the next goal becomes enabling optimized workflows. Many organizations have standardized their workflows so that they are already quite efficient. Moving from SAS® to R may mean the loss of existing tools like SAS® macros. This is common in areas such as table programming. This is another are that Atorus has aimed to help supplement R within clinical programming by releasing the open-source package Tplyr.

Tplyr offers organizations adopting R a tool for rapidly producing tables, making up ground that may be lost when transitioning languages. Tplyr aims to make the data preparation steps of common safety tables simple by breaking the problem down into basic components. Safety summaries often consist of two basic tasks: counting, or descriptive statistics. These can be combined in the same report, like a demographics table where you summarize numerous different variables of varying types. Tplyr allows you to stack these analyses together as different layers to the report. The package aims to keep all these concepts within a simple, readable, and intuitive syntax while still allowing a great deal of flexibility to the user. Tedious tasks such as string formatting of numbers are greatly simplified, allowing the user to focus on the report they are generating rather than the nuanced steps necessary to prepare it.

A key factor of Tplyr is that its focus is on the steps of data preparation and not on the cosmetic styling of the outputs. While it handles numeric formatting, the product of Tplyr is a data frame and not an HTML or RTF output. This is because sufficient R packages already exist to accomplish these tasks and could properly be leveraged alongside Tplyr.

ATORUS AND GSK JOINT OPEN-SOURCE PACKAGE DEVELOPMENT

To further open-source development for the clinical reporting pipeline, Atorus and GSK are working together. The goal of this partnership is to build, collect, and maintain a suite of R packages that support the conventional clinical reporting pipeline, much in the same fashion of the packages that Atorus has already released. The focus is to advance open-source solutions that meet the needs of the industry, rather than the needs specific to Atorus or GSK. This means that these solutions produced by this relationship serve a general purpose that should be reasonably adoptable by any organization.

The open-source philosophy of this engagement has a few guiding principles:

- If existing solutions are sufficient, leverage them where able.
- If an existing solution is close to what is required, contribute to that solution to enable it to meet requirements (if able).
- If no sufficient solution exists, then create a new one.
- Solutions should be modular rather than large black boxes, enabling combinations of different tools to meet a user or organization’s preference.

So far, this engagement has resulted in the production of three packages: timber, xportr, and metacore.

TIMBER

R is a different language than SAS® in many ways. One of the significant deviations that may be difficult for a SAS® programmer to grasp is the lack of a log file. R, being an interpreted language, does not have an immediate compliment to this. Furthermore, logging facilities in R tend to follow similar paradigms to those of other common programming languages – meaning that in most these tools the user specifies which information is reported in the log. The closest counterpart to the SAS® log in R is simply the console, which reports the code executed and any messages produced. But by its nature, R is not very verbose in comparison to SAS®.

The goal of the timber package is to offer a supplement for the log file when R is used within a clinical environment, but not replicate the SAS® log file in its entirety. SAS® logs are often useful to the user as a debugging tool, and thus tend to be filled with a great deal of information. This debugging information is not conventional in R, and R has plenty of tools specifically to debugging. The log produced by timber is

specifically targeted at producing information necessary for an audit trail. These logs collect important information, such as:

- Date/time of execution
- Program runtime
- Executing username
- Imported packages
- System information
- Namespace conflicts
- Generated errors or warnings
- Name of file being run as well as output file

This smaller and more isolated set of information allows for the log files to serve the purpose of an audit trail, and furthermore still enable the use automated tools to scan logs for any issues, such as unacceptable message – or in R issues like masked functions that could be problematic.

XPORTR

The generation of version 5 XPORT files in R is something that has been problematic within the clinical industry. Packages already exist for the reading and writing of XPORT files – including SASxport and the tidyverse package haven. The issue is that the files produced by these packages will not pass the necessary compliance checks for a regulatory submission.

One of the major problems goes down to the concept of variable lengths. Variable lengths do not exist within R in the same way that they exist in SAS®. In R, character strings will not be truncated (unless you happen to hit a memory limit). As this is not convention within R, both haven and SASxport do not allow for the manual setting of variable lengths when writing to an XPORT file. Furthermore, the defaults produced are not as expected by typical compliance checks. For example, if a variable is empty (meaning now row contains values), the length will be set to 0 rather than 1.

The development of the package xportr is great example of one of the principals of the Atorus and GSK open-source development partnership: if an existing solution is close to what is required, contribute to that solution to enable it to meet requirements (if able). The first stage of xportr development was to research both haven and SASxport to see what could be done to rectify the issue of setting variable lengths manually. The team established a solution and contributed this solution to the author of SASxport, and the solution was accepted (SASxport R Package). This then enabled SASxport to produce version 5 XPORT files that were capable of passing necessary compliance checks.

The next phase of development on xportr was to provide a simplified interface to the SASxport package. Users can provide the necessary metadata as a data frame or metacore object, and that metadata is appropriately set for use within SASxport. Furthermore, xportr runs a number of compliance checks the ensure compliance with regulatory standards, specific to that of the XPORT file. The xportr package also provides a functional messaging interface for the user that supplies information about the changes to the data frame along with different options of verbosity for these messages.

METACORE

The package metacore is the beginning of a foundation, focused on the use of metadata throughout different aspects of clinical statistical programming. The development of metacore split from that of xportr, where the team recognized that metadata can be leveraged throughout multiple stages of programming. When xportr writes out a dataset, metadata is necessary to identify what attributes should be set to a dataset and variable. For example, the dataset label, variable labels, and variable lengths all must be set. This information is typically readily available within dataset specifications – and is commonly used within SAS® programming processes as well.

Other aspects of SDTM and ADaM programming can also leverage metadata, be it in applying controlled terminology, identifying the sort sequence of a dataset, and more. Reporting programs can also benefit from the use of metadata, where code lists allow for the conversion of character to factor variables, which can be useful to supplement empty rows without the creation dummy datasets. Other possibilities include the creation of compliance and consistency checks, study deliverable quality and completeness checks, and more – as metadata can serve several different purposes.

The purpose of the package metacore is specifically to be a container of metadata. Organizations will inevitably handle their metadata in different ways, be it in Excel spreadsheets, SQL databases, or other formats. The schemas of these different sources of metadata will likely all be different – even when organizations are following CDISC standards. Packages designed for CDISC programming activities will need to leverage this metadata in different places. For example, xportr has an isolated purpose, and a package centered around SDTM programming techniques will have another purpose – but metadata can be leveraged within both packages.

The metacore package intends to create a standardized data structure that can be consumed by other packages able to leverage metadata. The xportr package can consume metadata from metacore object. As new packages are created, the intent is that they too can consume metadata from metacore, meaning that metadata only needs to be read once instead of potentially several times – and with a metacore object available less burden is placed on the user to provide different pieces of information.

The largest challenge of metacore is getting metadata from company specific formats to the standardized format of the container. Helper functions have been written to simplify the import of data from Excel spreadsheets, and readers have additionally been created for define.xml files. If these helpers are not sufficient, then fortunately the creation of a reader is a one-time activity that can then be reused moving forward – allowing users to still benefit from the standardized container provided by metacore.

CONCLUSION

The key aspect of the packages created by Atorus and GSK is that they are open-source. Anyone and any organization can download and leverage these packages. The question becomes what now? Where does the open-source development go from here? First – communication is key. The benefit of developing in the open is the collection of feedback from other users. If a requirement is not met, anyone is able to submit an issue via GitHub, where the development teams can respond, interact, and come up with a solution. Second, anyone can collaborate as well. Hosting the source code of a package on GitHub means that anyone can create their own copy of the source code, modify it, and submit it back to the original authors. Why submit an issue and wait for the update if capable developers can update the code themselves? The key here is that if an update is made – be sure to commit it back to advance the package for other users. Lastly, clinical open-source package development does not need to stop, and has not stopped, with Atorus and GSK. More and more organizations have been contributing to open-source, and the gaps in the clinical reporting pipeline continue to lessen in significance. Many of these programming activities serve no competitive advantage, as these are shared and universally faced challenges. All organizations stand to gain by having common, robust, open-source solutions freely available and adopted across industry.

REFERENCES

Chen, Ethan. 2020. “Electronic Submissions Update-From eCTD to CDISC Implementation and Beyond”, “*PharmaSUG Single Day Event – October 22-23, 2020*”, Available at https://www.fda.gov/media/143391/download

Tidyverse, 2019, “Welcome to the tidyverse”. Accessed April 1, 2021. https://www.tidyverse.org/

SASxport R Package, 2021, “PR #21: Updates for SAS length.” Accessed April 1, 2020 https://github.com/r-gregmisc/SASxport/pull/21

Your comments and questions are valued and encouraged. Contact the author at:

Mike Stackhouse
Atorus Research
Mike.Stackhouse@atorusresearch.com
https://www.atorusresearch.com

Nathan Kosiba
Atorus Research
Nathan.Kosiba@atorusresearch.com
https://www.atorusresearch.com