

Closing the Gap

Creating an End-to-End R Package Toolkit for the Clinical Reporting Pipeline

Eli Miller, Atorus & Ben Straub, GSK

Phuse Connect 2022

2022-04-19

Objective

Present and discuss three R packages, {metacore}, {timber}, {xportr}, and how they fit into the larger framework of the {pharmaverse} answer to the Clinical Reporting Pipeline.



Outline

Defining the Clinical Reporting Pipeline

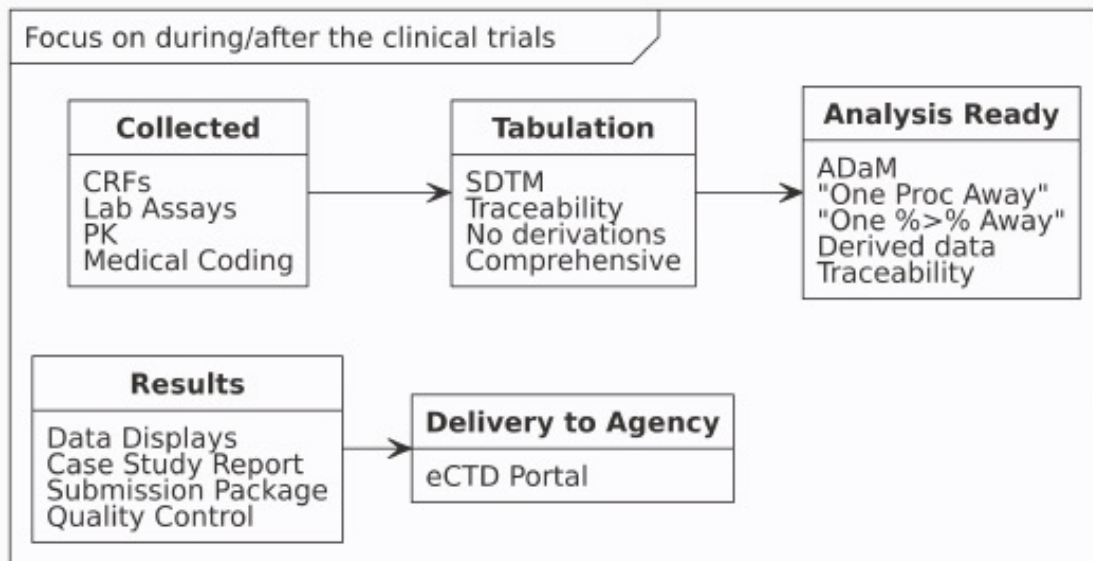
Open Source in a Regulated Environment

A collaboration begins between Atorus and GSK

Overview of the 3 Packages – {metacore}, {timber}, {xportr}

Welcome to the {pharmaverse}!

The Clinical Reporting Pipeline



Challenges of using an Open-Source Language in a Regulated Environment

An Incomplete List of Challenges

- SAS® processes have years of use within regulatory agencies and tend to take precedence.
- Company Firewalls and VPNs.
- Sharing of code and data.
- Does a R package bend to SAS® processes or does the package recreate/redefine the process?
- Who maintains the open source product?

An Incomplete List of Solutions

- Github, Gitlab and etc. help us to share, fix and build up code in an interactive and version-controlled process.
- R Consortium is working with the FDA to investigate delivering a submission in R.
- R packages already out there that mimic some of SAS® processes – {haven}, {SASxport}
- Pharma companies will have a vested interest in maintaining these packages.
- New graduates are coming out of school with R and Python skills

Risk Mitigation: Why xportr package is not on CRAN(yet)

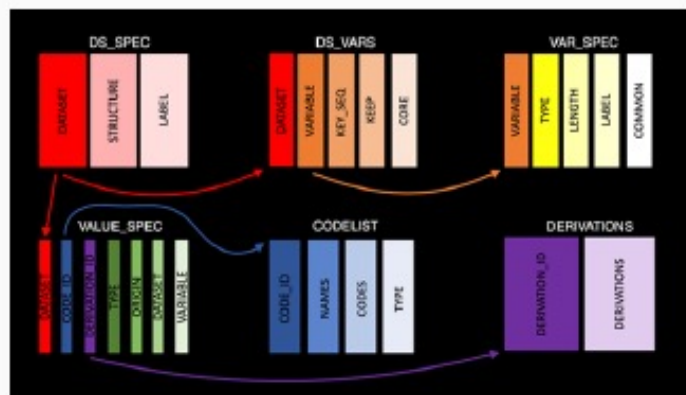
Backend dependancy issues

Limited time from OS developers

Difficulty Explaining the Need for Changes

- [Dec 11, 2020 - SASxport Issue Reported](#)
- [Feb 5, 2021 - PR to SASxport merged in](#)
- [Nov 17, 2021 - Issue with haven reported](#)
- [Nov 23, 2021 - Fix in underlying haven library patched](#)
- [Feb 21, 2022 - New release for haven backend](#)
- [Feb 25, 2022 - Backend updates made to haven](#)
- In Progress - Updates to xportr to switch to haven.

{metacore}



Challenges for R in CRP

- No available R packages that can ingest and parse metadata for clinical trial datasets or help build a define.xml.

Solutions

- Built functions for parsing xml data and other data types specific to clinical trial datasets.
- Functions and objects can be leveraged to help for dataset and variable creation as well as user for creating and codelist and derivations.

Future development path

- Working with different company's metadata files as well as in-development packages, e.g., admiral, to leverage metadata upstream or downstream.



```
-----
                        timber Metadata
-----
This log was generated using timber 0.0.0.9801
timber package version: 0.0.0.9801
timber license: MIT + file LICENSE
timber build: R 4.0.2; 2021-07-09 14:40:21 UTC; unix
timber link to repository: https://github.com/atorus-research/timber
-----
                        User Information
-----
User: daphne.laurel
-----
                        Session Information
-----
R version 4.0.2 (2020-08-22)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.5 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/libopenblas-p-r0.2.20.so

locale:
 [1] LC_CTYPE=C.UTF-8      LC_NUMERIC=C           LC_TIME=C.UTF-8       LC_COLLATE=C.UTF-8
 [5] LC_MONETARY=C.UTF-8  LC_MESSAGES=C.UTF-8   LC_PAPER=C.UTF-8      LC_NAME=C
 [9] LC_ADDRESS=C          LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
```

Challenges for R in CRP

- There is no SAS-like log file for R-based programs that show traceability and help to create audit-ready code.

Solutions

- Created a utilities and wrapper functions to capture user info, session information and warnings and errors in an easy-to-read format.
- Allows for recreation of script, not just the output.

Future development path

- Interactive display screen to batch submit multiple programs.
- What else should go in a log? Companies might have different needs.

{xportr}

```
adsl %>%  
  xportr_type(var_spec, "ADSL", "message") %>%  
  xportr_length(var_spec, "ADSL", "message") %>%  
  xportr_label(var_spec, "ADSL", "message") %>%  
  xportr_df_label(data_spec, "ADSL") %>%  
  xportr_write("adsl.xpt")
```

Challenges for R in CRP

- Writing a compliant xpt file using haven or other xpt-supported packages is challenging for types, lengths and labels

Solutions

- Adding well defined metadata
- Helps to write compliant xpt files
- Logs and messages for information about metadata
- Checks on the data before final output

Future development path

- How to ingest metadata specifications from different companies and different file formats

End to End Example

Q. What is the value of $\frac{1}{2}$?

Welcome to the {pharmaverse}!

