

Apply next-generation tools in real study

Jia Liu, Roche

ABSTRACT

Our real study focuses on chronic hepatitis B (CHB) virus infection. The programming team of this study is a mix of advanced R programmers and novices. The team explored using R based tools, OAK and Admiral, to support SDTMv and ADAM QC. Through the exploration and application, the team has gained lots of valuable experience on next-generation tools.

Roche is diligent to work on "next-generation" solutions for our data and analytics platforms to move towards increased automation. Next-generation tools OAK and Admiral are good examples. Admiral, toolbox for programming Clinical Data Interchange Standards Consortium (CDISC) compliant Analysis Data Model (ADaM) datasets in R. The OAK Garden aims to automate the clinical data flow from Data Collection to Data Tabulation thereby automating the study of SDTMv Datasets, SDTMv specifications, eSubmission Deliverables SDTM xpt files & SDTM Define.xml.

INTRODUCTION

With the advent of new technologies in data processing and analysis, Roche has been keeping up with the cutting edge of automation. Trying to create a set of automated data processing tools. The application of automation can not only effectively save human resources, improve efficiency, but also avoid some human errors. For clinical datasets, we are exploring to use open sourced R based tools 'Admiral' and 'OAK' to produce ADaM and SDTMv.

Admiral is aim to provide an open source, modularized toolbox that enables the pharmaceutical programming community to develop ADaM datasets in R. At the same time, OAK garden is aims at streamlining the clinical data flow from data collection to tabulation and the automation of the SDTMv datasets, specifications and "define.xml". Driven by metadata & global data standards, OAK Garden can automate ~80% SDTMv domains with ~22 reusable algorithms. The OAK garden comprises a number of components that will form the framework for simplifying and automating SDTM electronic submission deliverables.

ADMIRAL APPLIED IN REAL DATA TO PRODUCE ADAMS

As this is an early phase study, we do not have lots of complex programming design in ADAM datasets level. For now, the ADAM programming includes ADSL, ADEX, ADAE, ADVS and ADLB. It was not so easy at the first time, especially for the study specific derivation rules in lab dataset. However, it did get the efficiencies improved once the team got familiar with the R package 'Admiral'.

Take the ADSL dataset as example, Admiral provided a standard template, which is currently presented and tested using DM, EX, AE, LB and DS SDTM domains, to draft the ADSL dataset for users. In addition, users could make their own changes based that. The changes may be in two aspects. Firstly, some study specific additional variables derivation. Secondly, some provided variables derivation rules may not match with the study design. Therefore, users need to get a code review for the provided demo code and then decide whether to update the derivation rules for your own study.

The core task in the production is to code the derivation rules in a study specific way. All the packages that we need are {admiral}, {admiralroche} and {tidyverse}. This is a totally R based workflow.

The ADSL programming Flow includes:

1. Read in Data
2. Derive Period, Subperiod, and Phase Variables (e.g. APxxSDT, APxxEDT, ...)
3. Derive Treatment Variables (TRT0xP, TRT0xA)
4. Derive/Impute Numeric Treatment Date/Time and Duration (TRTSDT, TRTEDT, TRTDURD)

5. Derive Disposition Variables
6. Disposition Dates (e.g. EOSDT)
7. Disposition Status (e.g. EOSTT)
8. Disposition Reason(s) (e.g. DCSREAS, DCSREASP)
9. Randomization Date (RANDDT)
10. Derive Death Variables
11. Death Date (DTHDT)
12. Cause of Death (DTHCAUS)
13. Duration Relative to Death
14. Derive Last Known Date Alive (LSTALVDT)
15. Derive Groupings and Populations
16. Grouping (e.g. AGEGR1 or REGION1)
17. Population Flags (e.g. SAFFL)
18. Derive Other Variables
19. Add Labels and Attributes

For the above derivations, the most interesting one is date time imputation related derivation, which is also the most complex one. Package Admiral has also provided functions to help users to do their own derivations on that, like `derive_vars_dt()`, `derive_vars_dtm()`, and `derive_vars_dtm_to_dt()`. Users could control the reference date time and imputation rules by parameters.

For example, impute start and end time of exposure to first and last respectively without imputing date.

```
ex_ext <- ex %>%
  derive_vars_dtm(
    dtc = EXSTDTC,
    new_vars_prefix = "EXST"
  ) %>%
  derive_vars_dtm(
    dtc = EXENDTC,
    new_vars_prefix = "EXEN",
    time_imputation = "last"
  )
```

For more details in the demo code, please find in the [pharmaverse.github](https://github.com/pharmaverse/admiral) of Admiral.

ADVANTAGES AND THINGS NEED IMPROVED

Using Admiral to produce ADAMs does have many advantages; especially lots of new employees come to industry with solid R programming experience.

However, of course some things should be improved in Admiral. For example, in ADLB programming, Admiral has not provided enough metadata that needed in the production. In version 0.1.1, Admiral provided lab-grading metadata NCI-CTCAEv4 and NCI-CTCAEv5. However, DAIDSV21 is required for this real study. Therefore, the study team can only try to mockup the grading metadata by hand which is really risky and time-consuming. Nevertheless, according to the development team of Admiral, they are working on the rest of metadata to make them alive. If more metadata could be provided in the near future, Admiral could support more and more studies.

OAK APPLIED IN REAL DATA TO PRODUCE SDTM

For our real study, we have 32 domains in total. Some of them are study specific domains. So, for those study specific, without standard mapping available, domains, we haven't tried to use OAK garden to produce. However, for those standard and common domains, they all could be produced by OAK.

The core concept of OAK Garden is the reusable algorithms. SDTMv mappings are implemented as algorithms that transform the collected source (eCRF, non-CRF) data into the target tabulation data model. About 22 unique mapping algorithms have been identified to perform the data transformation. The algorithms can be re-used across domains. They are pre-defined for data standards in the Pistil semantic model in GDSR. The user can also reuse the algorithms to map the non-standards in the study using MINT+.

The workflow for the real data to apply OAK is:

1. Prepare spec for SDTMv according to the study design
2. Prepare Study Configuration Files, including Timepoint configuration, Reference date configuration, Epoch configuration
3. Set environment variables SAFFRON_API_URL for the MINT+
4. View all available specification versions in saffron
5. Load the study SDTMv eCRF mappings metadata and controlled terminology from saffron
6. Load CRF/non-CRF data and GDSR meta into oak_pkg_env
7. Create SDTMv Domains Using OAK. e.g. `create_domain("DM", study_sdtmv_metadata, study_cont_terms)`

ADVANTAGES AND THINGS NEED IMPROVED

OAK garden is a SDTM production related packages' combination. All members inside of it are aiming to automate the process of production. As SDTM production automation is a huge scope, it takes time to make all automation alive. From this journey, we can find that some steps are not as expected as automatic. However, as far as we know, the development team is trying their best to deliver more and more tools to support the OAK garden. For example, the updates of mint+ to use one line of code in R program to synchronize between R environment and mint+ by study ID. It saved lots of time for users to prepare the setup.

However, as far as we can see in this real study, some mapping criterial is not as expected in default functions provided by the package. And it's not so easy for user to update the mapping rules, as all domains are supposed to be created by a unified function 'create_domain()'.

Besides that, users should prepare Study Configuration Files one by one by themselves. In our real study, we are doing it by hand, more specific, by inputting csv file and upload to R environment. It is expected to be more automatically.

CONCLUSION

It will make things much easier if R could be used in the production. As open sourced R packages update much faster than traditional tools, which means the most effective functions can be applied in real time. Besides that, new tools and packages burst forth every day on R, which means we can join the wave anytime. The best practice for ADAM and SDTMv programming in R based tools, powerful packages, Admiral and OAK.

References

Website Admiral <https://pharmaverse.github.io/admiral/cran-release/index.html>.

Website OAK <https://botany.pages.roche.com/oak/index.html>

Article in conference proceedings WHO. Evidence to recommendations table and GRADE table, Need for a hepatitis B vaccine booster dose following primary immunization.

Website Stream user manual. https://pages.github.roche.com/MDIS/stream_doc/um/techdoc/toc_vads.html

CONTACT INFORMATION

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

<Jia Liu>

<Roche>

<+86 13956158028>

<jia.liu.jl12@roche.com>