Paper OS20

Advancing TFL Automation: R-Based Meta-Programming Powered by ARS

Malan Bosman, Clymb Clinical, Bloemfontein, South Africa

ABSTRACT

In the industry's steady move towards TFL automation, there is still a considerable gap when it comes to automating Analysis Results. With the recent development of Analysis Results Standard (ARS, launched by CDISC in April 2024), the opportunity of building TFL automation tools on industry standards was introduced. *siera* is an R package that ingests ARS metadata and generates fully functional R code – code that can be run as-is to produce the Analysis Results Dataset (ARD). The ARD is a single dataset that contains all the results required in the final TFLs described in the ARS. Far from being a "black box" producing the ARD, this R package instead provides the code, making it inspectable and changeable. It brings the user one step closer to automating TFLs and ultimately plays an important role in end-to-end TFL automation – all while utilizing industry standards (ARS) and Open-Source technology (R).

INTRODUCTION

The vision of automating analysis results using metadata has been steadily taking shape in the clinical research industry. Much of the groundwork has been laid in recent years, through initiatives like CDISC 360, and the development of the CDISC Analysis Result Standard (ARS). The vision set forth is clear and promising in terms of time and cost saving, as well as increased quality of analysis results generation. Such a potential workflow, incorporating ARS and envisioning automated Analysis Programs, was presented as part of the eTFL Portal initiative [1].

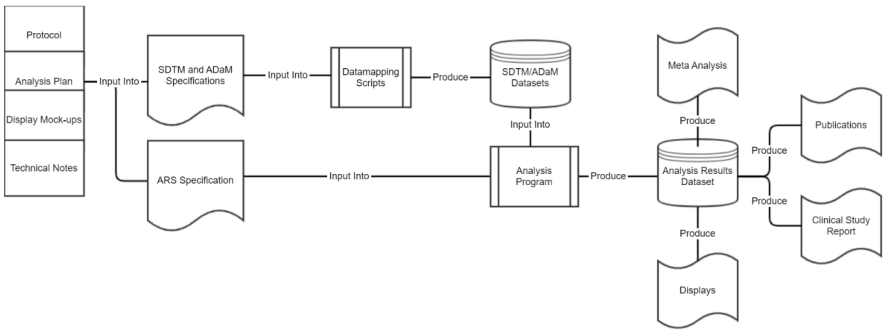

Figure 1- Potential Workflow using ARS and TFL automation

The "siera" R package fits into Figure 1 as an automation engine, or "Analysis Program", ingesting ARS metadata and producing Analysis Results Datasets (ARDs). Strictly speaking, *siera* does not directly produce the ARDs, but rather R scripts that can be run as-is to produce ARDs when ingesting ADaM datasets. Before discussing the inner workings of *siera,* however, it is worth having a brief overview of the foundation for *siera*, namely the ARS.

Analysis Results Standard

The Analysis Results Standard (ARS) was created by CDISC with the "aim to facilitate automation, reproducibility, reusability, and traceability of analysis results data". This addresses the inefficiency of having no standard way of describing and organizing clinical trial analysis results found in tables and figures. The ARS therefore provides a Logical Data Model that describes analysis results and associated metadata. [2]

The ARS contains a Logical Data Model that describes analysis results and the associated metadata. This model is represented by a schema, showing all components with accompanying metadata linked to a single reporting event. This metadata is divided into model components, which together describe the transformations that need to be applied to ADaM dataset(s) to produce the results for the reporting event. These results are combined in a flat table with context for each result, which is known as an Analysis Results Dataset (ARD).

To create a result (and by implication many results to create an ARD), various model components need to be considered in combination. For the purposes of this paper, a handful of key components will be discussed, namely Analysis Sets, Data Subsets, Analysis Groupings, and Analysis Methods. Each analysis contains an Analysis Set, the subject population, Data Subset, conditions describing the selection of data to be included in the analysis, Analysis Grouping, characteristics used to cluster the subject populations and Analysis Method, the statistical operation applied to the data. The metadata for the reporting event is organized into classes with fields that house the metadata elements as shown in Figure 2:

Deleted: ¶

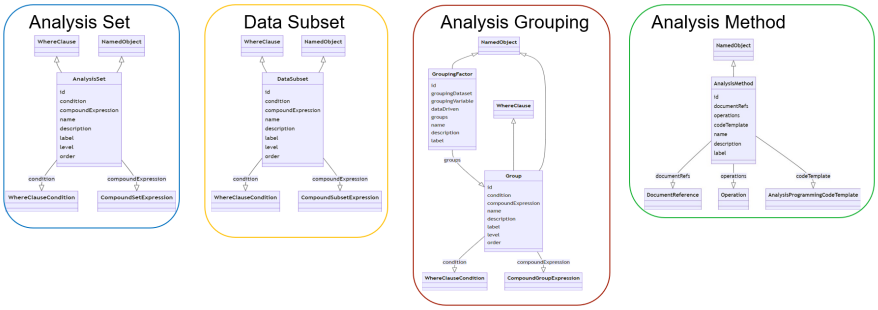

Figure 2 - ARS Logical Model Classes and their Metadata Components

These four key model components are explicitly used in the siera package. For each output, all analyses linked to the output are looped through its Analysis Set, Data Subset, Analysis Grouping and Analysis Method. By doing so, results are obtained for each analysis, all of which are combined to form the ARD.

SIERA: A META-PROGRAMMING APPROACH TO AUTOMATION

Building on ARS

With ARS metadata available for a reporting event, an automation tool / engine is required to ingest this metadata and produce the results. However, simply operating as a "black box" is not a desired state for such an automation engine. Instead of directly generating analysis results, it would be preferable to have an "intermediate" step – i.e. seeing the code / scripts that are run in order to produce the analysis results. With *siera*, then, ARS metadata is ingested, and ready-to-run R scripts are generated, which can be run with the appropriate ADaM dataset to produce the ARDs. These R scripts are therefore meta-programmed, using the metadata from ARS. This workflow is visualized below in Figure 3:

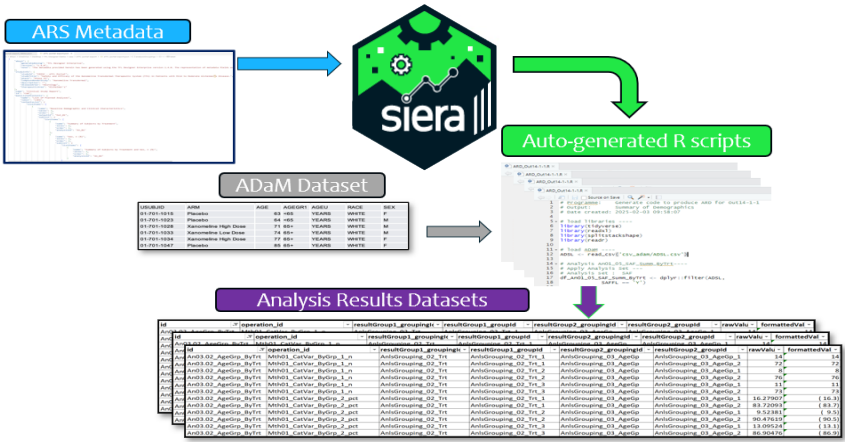

Figure 3 -siera workflow: ARS to ARD

siera in combination with other software

Naturally, software exists in an ecosystem with other tools, each fulfilling specific purposes. Two direct links for *siera* are required in order to have a complete process, from design (e.g. shell creation) to execution (e.g. TFL Displays).

First, the ARS metadata needs to be populated for a specific reporting event (e.g. CSR). To populate this metadata, information from SAP, Protocol, TFL Shells etc. is required, and needs to be combined in a structured, logical way, complying with the ARS Logical Data Model. Open-Source software fulfilling this purpose is TFL Designer Community Version, where shells can be designed with appropriate metadata, which can be exported as ARS metadata in JSON or Excel format [3].

Second, once the ARDs have been created by *siera,* the final step would usually include producing displays (TFLs). To produce this, Analysis Display Metadata (ADM) is needed, which describes the formatting of the final displays. The TFL Designer (Enterprise) exports ADM, which can be ingested programmatically to combine the ARD with ADM, producing displays.

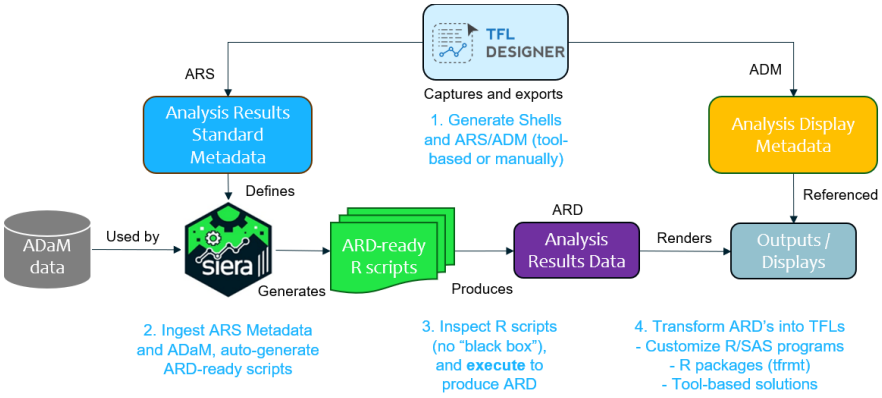

Figure 4 - siera in combination with other software solutions

Getting started with *siera*

To use siera, a fully completed ARS metadata file for a reporting event is used as input. This can be either JSON (the official ARS format), or the adapted ARS metadata in Excel (as used in CDISC's ARS v1 example) [4]. Snippets of this metadata can be seen in Figure 5:

JSON

Excel

	id	version	name
2	An_01	1	Summary of Subjects by Treatment
3	An_02	1	Summary of Subjects by Treatment and Sex, n (%)
4	An_02_Total	1	Summary of Subjects by Sex, n (%)
5	An_03	1	Summary of Age, Years by Treatment
7	An_03_Total	1	Summary of Age, Years
7	An_04	1	Summary of Subjects by Treatment and Age groups (years), n (%)
8	An_04_Total	1	Summary of Subjects by Age groups (years), n (%)
9	An_05	1	Summary of Subjects by Treatment and Race, n (%)
10	An_05_Total	1	Summary of Subjects by Race, n (%)
11	An_06	1	Summary of Subjects by Treatment and Ethnicity, n (%)
12	An_06_Total	1	Summary of Subjects by Ethnicity, n (%)
13	An_07	1	Summary of Subjects by Treatment and Country of participation, n (%)
14	An_07_Total	1	Summary of Subjects by Country of participation, n (%)
15	An_08	1	Summary of Baseline Weight (kg) by Treatment
16	An_08_Total	1	Summary of Baseline Weight (kg)
17	An_09	1	Summary of Baseline Height (cm) by Treatment
18	An_09_Total	1	Summary of Baseline Height (cm)
19	An_10	1	Summary of Baseline BMI (kg/m^2) by Treatment
20	An_10_Total	1	Summary of Baseline BMI (kg/m^2)
21	An_11	1	Summary of Duration of Disease (Months) by Treatment
22	An_11_Total	1	Summary of Duration of Disease (Months)

Figure 5 - ARS metadata snippets. JSON and Excel versions

Once the R scripts are produced, ADaM datasets as applicable need to be used as input to these R scripts, so the data subsets, grouping etc. can apply be applied to the ADaM data. The direct output of running siera is meta-programmed R scripts, ready to be run by the user to produce ARDs. These scripts are of course transparent and can be updated by the owner as needed/allow

Below is a quick guide using and example to getting started with siera:

1. Installing the package can be done with:

```
# install.packages("siera")
```

2. Running Siera, using an ARS JSON file:

```
library(siera)


# 1. the ARS JSON File: An example is included in ARS_example function

json_path <- ARS_example("ARS_V1_Common_Safety_Displays.json")

# 2. store generated ARD scripts: Update tempdir() to desired location

# A suggestion would be to use "getwd()" for outputting to current location

output_folder <- tempdir()

#3.  ADaM: Use Example ADaM from ARS_example function.

# We use dirname(normalizePath()) to specify the folder containing ADaMs

ADaM_folder <- dirname(normalizePath(ARS_example("ADSL.csv")))

# run the readARS function with these 3 parameters.

readARS(json_path, output_folder, ADaM_folder)
```

3. The result of the above code is multiple R programmes (1 for each output to be generated). Each of these R scripts now also contains a reference to ADaM dataset(s), and can be run to produce ARD. Figure 6 shows the result of running the "readARS()" function: Separate R scripts are produced, each set up to produce an ARD for the applicable output.

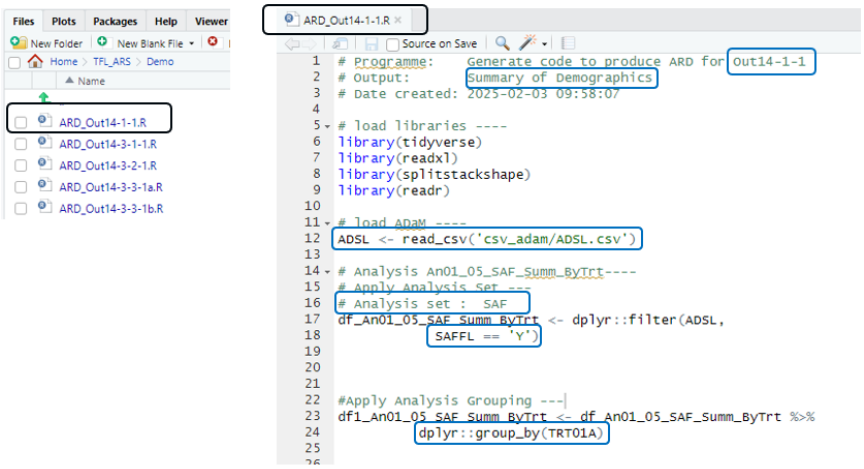

Figure 6 - Result of the readARS call in siera: R scripts meta-programmed

As can be seen in Figure 6, the produced R script is populated with metadata, as read in from the ARS JSON file. Examples in the screenshot include the Analysis Set being applied in R code, "*SAFFL = "Y"*", as well as the grouping of the ADSL dataset by TRT01A variable (*group_by(TRT01A)*). By continuing to translate these metadata elements into workable R code, the script to generate all results for an output is created. These results are finally combined in the same script, producing the ARD for a specific output.

4. Finally, the produced R program can be run as-is (it is advised to confirm if the location of the ADaM dataset is correct on the local system first).

```
# location to one of the created R scripts:

ARD_program <- file.path(paste0(output_folder,"/ARD_Out14-1-1.R"))

source(ARD_program)
```

The result of this command is that an ARD with already formatted results is generated.

Considerations for implementing *siera*

Using Siera and ARS metadata to produce R scripts and ARDs is a change from traditional TFL generation. The following needs to be considered in implementing Siera as part of an automated workflow:

- R infrastructure, e.g. an R Studio Workspace, knowledgeable R resources, etc. needs to be in place
- A strong knowledge of ARS metadata and Logical Data Model would help the team member responsible for setting the ARS metadata up.
- A team composition (e.g. amount of Statisticians and Programmers) may need to be updated, as setting up ARS metadata requires more involvement in Programming Setup, but since automation is applied, less direct programming would likely be needed (rather, Programmers can become owners of the R scripts producing ARDs)

CONCLUSION

Impact

Implementing siera into workflows would mean a larger focus on metadata setup, and a lesser focus on manual programming. Getting the ARS metadata in good shape would take a focused effort by Biostats members, and would likely take up more time than metadata setup takes in current workflows. However, this effort would be more than rewarded by the decrease in manual programming efforts, making use of the automation capabilities of ARS and siera.

A second impact would be that the time spent on repetitive programming tasks would be reduced, freeing resources up to spend more time on complex tasks, such as configuration of statistical procedures, etc. This allows for a more productive use of resources, automating repetitive programming with automation using ARS and siera.

As a result of automation, the overall cost of programming would decrease, while quality, consistency and efficiency would increase. Ultimately, this is aligned with the goal of the industry, producing more reliable results at less cost, benefiting all stakeholders and the patient in the end.

Learnings

Another learning has been that the use of ARDs in the workflow of TFL generation requires major shift in the status quo. Traditionally, ADaM datasets are programmed directly into RTF or PDF displays, without a step where Analysis Results Datasets are created with machine-readable data, connected to ARM. This change needs to be designed carefully, and the benefits understood well to have successful transitions.

From interactions with stakeholders, a recurring piece of advice was that the idea of passing metadata (and ADaM datasets) into a “black box” which provides the results, is not a comfortable proposition for most. The conversion process needs to be transparent, changeable, and understandable. With siera, this has been built into the design, to provide R scripts rather than results directly.

The user of siera and automation software each has a unique context. To accommodate this, the package needs to provide flexibility in usage. Going forward, this will be considered in feature designs, where an emphasis will be placed on allowing the user to, e.g. provide code snippets for operations (e.g. p-value calculation), rather than being restricted to a specific built-in set of operations within siera.

Next steps

The next steps for siera would be to have implementations into existing workflows. This way, updated functionality can be explored for the siera package. The valued feedback from actual users would lead to more updates and upgrades, which in turn would enhance automation for Biostats teams. Users can obtain the package from CRAN, start to use the examples shipped with the package, and reach out via email or Github with questions and suggestions.

REFERENCES

[1] B. B. Bess LeRoy. [Online].

[2] CDISC, “Analysis Result Standard,” April 2024. [Online]. Available: https://www.cdisc.org/standards/foundational/analysis-results-standard.

[3] Clymb Clinical, [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://clymbclinical.com/wp-content/uploads/2024/09/TFL-Designer-Community-Release-Webinar-Slides-15th-August-1.pdf.

[4] CDISC, “ARS V1,” April 2024. [Online]. Available: https://github.com/cdisc-org/analysis-results-standard/tree/main/workfiles/examples/ARS%20v1.

ACKNOWLEDGEMENTS

I would like to acknowledge Bhavin Busa (Co-Founder of Clymb Clinical and Product Owner of CDISC’s ARS) for his continued guidance and support on this project. Furthermore, I would like to acknowledge Richard Marshall (Principal Architecture of ARS model) for his technical insights and assistance with understanding key concepts.

RECOMMENTED READING / WATCHING

- eTFL Portal utilizing ARS: https://www.cdisc.org/events/webinar/introducing-new-cdisc-etfl-portal
- ARS Public Review: https://www.cdisc.org/events/webinar/analysis-results-standard-public-review

CONTACT INFORMATION

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

Author Name: Malan Bosman

Company: Clymb Clinical

Email: mbosman@clymbclinical.com

Website: https://clymbclinical.com/

Brand and product names are trademarks of their respective companies.