Streamlining Clinical Trial Analysis and Reporting: An Innovative Approach using R for Interactive Reporting

Benjamin Wang
Merck & Co., Inc., Rahway, NJ, USA
2025-07-09

Agenda

- Introduction
- Metadata Approach to Generate Outputs
- Interactive DMC report
- Summary

Introduction

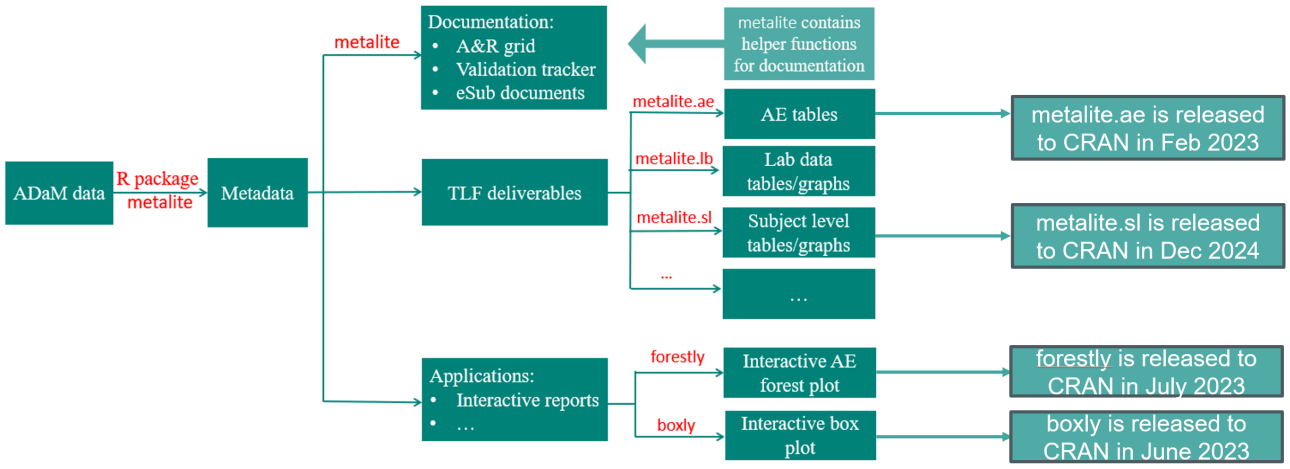
Problem Statement

- Large trials generate hundreds of analysis result deliverables in the form of static Tables, Listings & Figures (TLFs).
- Static reports can be lengthy and challenging for stakeholders to review.
- Additional tables and listings might be required for review and interpretation of results.

Requirements

- Our use case is specifically designed for Data Monitoring Committee (DMC) review of TLFs.
- The interactive reports need to be shared with other departments internally or with the external stakeholders .
- End-to-end process and tools to generate interactive reports from ADaM data should be developed following internal A&R workflow.
- The interactive reports should contain same information as in static TLFs.
- Cross validation should be conducted for the results generated from interactive reporting tools with static output.

Workflow


ADaM data
R package
metalite
Metadata
metalite
Documentation:
• A&R grid
• Validation tracker
• eSub documents
metalite contains helper functions for documentation
TLF deliverables
metalite.ae
AE tables
metalite.lb
Lab data tables/graphs
metalite.sl
Subject level tables/graphs
...
…
metalite.ae is released to CRAN in Feb 2023
metalite.sl is released to CRAN in Dec 2024
Applications:
• Interactive reports
• …
forestly
Interactive AE forest plot
boxly
Interactive box plot
forestly is released to CRAN in July 2023
boxly is released to CRAN in June 2023

Metadata Approach to Generate Reports

Metadata

What is metadata?

- Metadata is data about data that provides context to understand data elements
- It consists of sets of attributes that provide additional details on the data, variables, and formatting that are used to create TLFs.

Benefits of using Metadata:

- Helps to define attributes once and use it multiple times.
- The individual properties of an element are inherited rather than duplicated.
- It increases efficiency, traceability and decrease manual error.

metalite

metalite is an R package designed to create metadata structure for ADaM data analysis and reporting.

metalite offers a foundation to simplify tool development and create standard workflows.

Principles of metalite package:

- Automation: prefer a function call more than a checklist.
- Single-entry: enter in one place, sync to all deliverables.
- End-to-end: follow all steps in SOPs from define, develop and validate.

Workflow

Example of unified metadata generated by metalite:

```
meta
```

```
## ADaM metadata:
##    .$data_population     Population data with 254 subjects
##    .$data_observation    Observation data with 1191 records
##    .$plan    Analysis plan with 4 plans
##
##
##   Analysis population type:
##     name        id  group var       subset                          label
## 1 'apat' 'USUBJID' 'TRTA'     SAFFL == 'Y' 'All Participants as Treated'
##                    group_order
## 1 Xanomeline High Dose, Placebo
##
##
##   Analysis observation type:
##     name        id  group          var       subset           label    ae_var
## 1 'wk12' 'USUBJID' 'TRTA' AEREL, AESER SAFFL == 'Y' 'Weeks 0 to 12' 'AEDECOD'
##
##
##   Analysis parameter type:
```

Interactive DMC Report

Solution

- The entire solution is built using the `metalite` framework.
- Interactive features leverage R packages `crosstalk`, `reactable`, `plotly` and JavaScript snippet.
- The packages `forestly`, `boxly`, and `metalite.sl` were developed to encapsulate these interactive components.
- R Markdown is used to generate the final deliverable as HTML output.

Output

Example output from our packages:

AE Summary table from `metalite.ae`:

Summary of Adverse Events
Weeks 0 to 12
All Participants as Treated

	Placebo		Low Dose		High Dose		Total	
	n	(%)	n	(%)	n	(%)	n	(%)
Participants in population	86		84		84		254	
with one or more adverse events	69	(80.2)	77	(91.7)	79	(94.0)	225	(88.6)
with no adverse events	17	(19.8)	7	(8.3)	5	(6.0)	29	(11.4)
with drug-related[a] adverse events	44	(51.2)	73	(86.9)	70	(83.3)	187	(73.6)
with serious adverse events	0	(0.0)	1	(1.2)	2	(2.4)	3	(1.2)

[a]Determined by the investigator to be related to the drug.

Source: [CDISCpilot: adam-adsl; adae]

Interactive forest plot from `forestly`:

AE Criteria

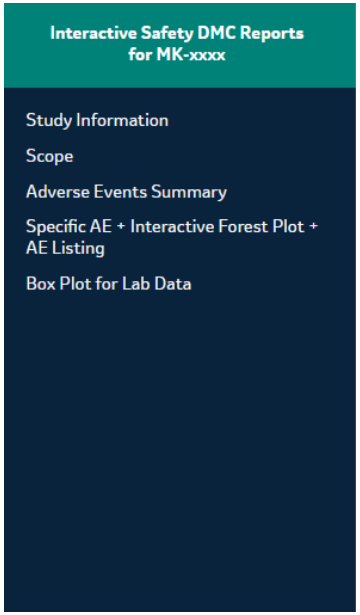
any adverse events

Incidence (%) in One or More Treatment Groups

0 30

0 3 6 9 12 15 18 21 24 27 30

Demo



Interactive Safety DMC Reports for MK-xxxx

Study Information

This report is limited to adverse events analysis for POC purposes. Please refer to the assembled DMC report in the word document for other DMC analysis results.

Scope

- AE summary
- AE specific
- AE forest plot
- AE listing
- Lab box plot

Demo: https://rconnect.merck.com/interactive-dmc-report-demo-20250410/

Summary

Summary

For analysis and reporting in clinical trial development, the metalite framework provides an unified approach to:

- Enable accuracy, traceability, and automation of the deliverables.
- Enable automation of A&R grid and some sections of ADRG.
- Build end-to-end process to generate TLFs, especially for interactive reports from ADaM data.
- Can be extended to allow creation of CDISC Analysis Results Standards.

Advantages of interactive DMC reports:

- Elegant solution to integrate a large volume of information from TLFs into one interactive plot.
- Output is a standalone HTML file without a backend server and can be easily shared.

Reference

Everything is available on CRAN and GitHub!

- **Installation:**

```
devtools::install("metalite")
devtools::install("metalite.ae")
devtools::install("metalite.sl")
devtools::install("forestly")
devtools::install("boxly")
```

- **Documentation:**
  - `metalite`: https://merck.github.io/metalite/
  - `metalite.ae`: https://merck.github.io/metalite.ae/
  - `metalite.sl`: https://merck.github.io/metalite.sl/
  - `forestly`: https://merck.github.io/forestly/
  - `boxly`: https://merck.github.io/boxly/

Team Members

Yilong Zhang	Ben Wang
Yujie Zhao	Hiroaki Fukuda
Nan Xiao	Sarad Nepal
Yulia Sidi	Ginnaram Madhusudan
PoYao Niu	Jeetender Chauhan
Brian Lang	Chen Wang
Xuan Deng	Li Ma
Bing Liu	Shuo Geng
Jung-Jin Lee	Narendra Kolisetty

Also particular thanks to Keaven Anderson, Abhilash Chimbirithy, Sandeep Meesala, Suhas Sanjee for project alignment and planning.

Thank you

Email: benjamin.wang@merck.com