Paper PP31

{sdtm.oak} – SDTM programming in R

Rammprasad Ganapathy, Roche/Genentech, San Francisco, USA
Shiyu Chen, Atorus Research, Hillsboro, USA

ABSTRACT

{sdtm.oak} is an EDC (Electronic Data Capture systems) and Data Standard agnostic solution that enables the pharmaceutical programming community to develop CDISC SDTM datasets in R. The reusable algorithms concept in {sdtm.oak} provides a framework for modular programming and also can potentially automate SDTM creation based on the standard SDTM spec.

INTRODUCTION

{sdtm.oak} v0.1 is now available on CRAN. In this paper, we will introduce the package and showcase its features and present a roadmap for future releases. {sdtm.oak} is developed in collaboration with volunteers from several companies, including Roche, Pfizer, GSK, Pattern Institute, Transition Technologies Science, and Atorus Research. {sdtm.oak} is also sponsored by CDISC COSA with a vision of being part of CDISC 360 to address end-to-end standards development and implementation.

IN THIS RELEASE

The v0.1 release of {sdtm.oak} users can create the majority of the SDTM domains. Domains that are NOT in scope for the v0.1.0 release are DM (Demographics), Trial Design Domains, SV (Subject Visits), SE (Subject Elements), RELREC (Related Records), Associated Person domains, creation of SUPP domain, and EPOCH variable across all domains.

FILLING THE GAP

{sdtm.oak} package addresses a critical gap in the pharmaverse suite by enabling study programmers to create SDTM datasets in R, complementing the existing capabilities for ADaM, TLGs, eSubmission, etc.

Let's explore the challenges with SDTM programming. Although SDTM is simpler with less complex derivations compared to ADaM, it presents unique challenges. Unlike ADaM, which uses SDTM datasets as its source with a well-defined structure, SDTM relies on raw datasets as input. These raw datasets can vary widely in structure, depending on the data collection and EDC (Electronic Data Capture) system used. Even the same eCRF (electronic Case Report Form), when designed in different EDC systems, can produce raw datasets with different structures.

Another challenge is the variability in data collection standards. Although CDISC has established CDASH data collection standards, many pharmaceutical companies have their own standards, which can differ significantly from CDASH. Additionally, since CDASH is not mandated by the FDA, sponsors can choose the data collection standards that best fit their needs.

There are hundreds of EDC systems available in the marketplace, and the data collection standards vary significantly. Creating a single open-source package to work with all sorts of raw data formats and data collection standards seemed impossible. But here's the good news: not anymore! The {sdtm.oak} team has a solution to address this challenge.

{sdtm.oak} is designed to be highly versatile, accommodating varying raw data structures from different EDC systems and external vendors. Moreover, {sdtm.oak} is data standards agnostic, meaning it supports both CDISC-defined data collection standards (CDASH) and various proprietary data collection standards defined by pharmaceutical companies. The reusable algorithms concept in {sdtm.oak} provides a framework for modular programming, making it a valuable addition to the pharmaverse ecosystem.

EDC & DATA STANDARDS AGNOSTIC

We adopted the following innovative approach to make {sdtm.oak} adaptable to various EDC systems and data collection standards:

- SDTM mappings are categorized as algorithms and developed as R functions.
- Used datasets and variables are specified as arguments to function calls.

CORE CONCEPT – REUSABLE ALGORITHMS

The SDTM mappings that transform the collected source data (eDT: External Data Transfer) into the target SDTM data model are grouped into algorithms. These mapping algorithms form the backbone of {sdtm.oak}.

Key Points:

- Algorithms can be re-used across multiple SDTM domains.
- Programming language agnostic: This concept does not rely on a specific programming language for implementation.
- Automation-ready: Algorithms can be pre-specified for data collection standards in metadata repository (MDR).

The {sdtm.oak} package includes R functions to handle these algorithms. Some of the basic algorithms are below, also explaining how these algorithms can be used across multiple domains.

Algorithm Name	**Description**	**Example**
assign_no_ct	One-to-one mapping between the raw source and a target SDTM variable that has no controlled terminology restrictions. Just a simple assignment statement.	MH.MHTERM AE.AETERM
assign_ct	One-to-one mapping between the raw source and a target SDTM variable that is subject to controlled terminology restrictions. A simple assign statement and applying controlled terminology. This will be used only if the SDTM variable has an associated controlled terminology.	VS.VSPOS VS.VSLAT
assign_datetime	One-to-one mapping between the raw source and a target that involves mapping a Date or time or datetime component. This mapping algorithm also takes care of handling unknown dates and converting them into. ISO8601 format.	MH.MHSTDTC AE.AEENDTC
hardcode_ct	Mapping a hardcoded value to a target SDTM variable that is subject to terminology restrictions. This will be used only if the SDTM variable has an associated controlled terminology.	MH.MHPRESP = 'Y' VS.VSTEST = 'Systolic Blood Pressure' VS.VSORRESU = 'mmHg'
hardcode_no_ct	Mapping a hardcoded value to a target SDTM variable that has no terminology restrictions.	FA.FASCAT = 'COVID-19 PROBABLE CASE' CM.CMTRT = 'FLUIDS'
condition_add	Algorithm that is used to filter the source data and/or target domain based on a condition. The mapping will be applied only if the condition is met. This algorithm has to be used in conjunction with other algorithms, that is if the condition is met perform the mapping using algorithms like assign_ct, assign_no_ct, hardcode_ct, hardcode_no_ct, assign_datetime.	If MDPRIOR == 1 then CM.CMSTRTPT = 'BEFORE'. VS.VSMETHOD when VSTESTCD = 'TEMP' If collected value in raw variable DOS is numeric then CM.CMDOSE If collected value in raw variable MOD is different to CMTRT then map to CM.CMMODIFY

Here is an example of reusing an algorithm across multiple domains, variables, and also to a non-standard mapping:

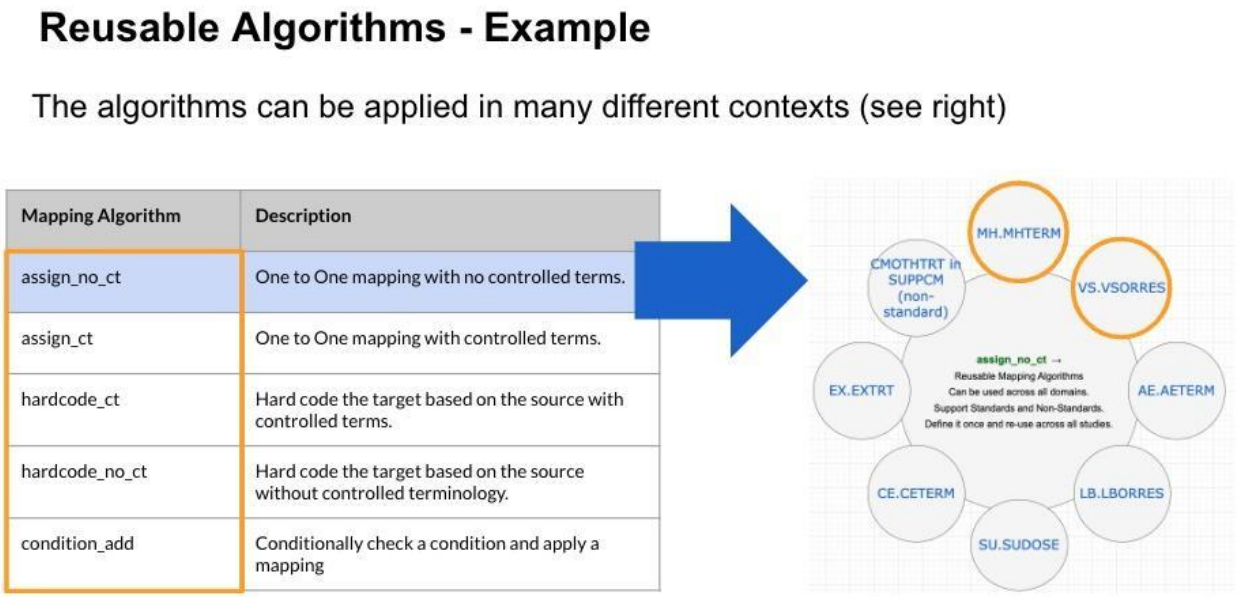

Fig.1. Example of reusable algorithms.

FUNCTIONS AND ARGUMENTS

All the aforementioned algorithms are implemented as R functions, each accepting the raw dataset, raw variable, target SDTM dataset, and target SDTM variable as parameters.

```
library(sdtm.oak)
library(dplyr)

cm_raw <- tibble::tribble(
  ~oak_id, ~raw_source, ~patient_number, ~MDRAW, ~DOSU, ~MDPRIOR,
  1L, "cm_raw", 375L, "BABY ASPIRIN", "mg", 1L,
  2L, "cm_raw", 375L, "CORTISPORIN", "Gram", 0L,
  3L, "cm_raw", 376L, "ASPIRIN", NA, 0L
)

study_ct <- tibble::tribble(
  ~codelist_code, ~term_code, ~term_value, ~collected_value, ~term_preferred_term,
~term_synonyms,
  "C71620", "C25613", "%", "%", "Percentage", "Percentage",
  "C71620", "C28253", "mg", "mg", "Milligram", "Milligram",
  "C71620", "C48155", "g", "g", "Gram", "Gram"
)

cm <-
  # Derive topic variable
  # SDTM Mapping - Map the collected value to CM. CMTRT
  assign_no_ct(
    raw_dat = cm_raw,
    raw_var = "MDRAW",
    tgt_var = "CMTRT"
  ) %>%
  # Derive qualifier CMDOSU
  # SDTM Mapping - Map the collected value to CM. CMDOSU
  assign_ct(
    raw_dat = cm_raw,
    raw_var = "DOSU",
    tgt_var = "CMDOSU",
```

```
    ct_spec = study_ct,
    ct_clst = "C71620",
    id_vars = oak_id_vars()
  ) %>%
  # Derive qualifier CMSTTPT
  # SDTM mapping - If MDPRIOR == 1 then CM.CMSTTPT = 'SCREENING'
  hardcode_no_ct(
    raw_dat = condition_add(cm_raw, MDPRIOR == "1"),
    raw_var = "MDPRIOR",
    tgt_var = "CMSTTPT",
    tgt_val = "SCREENING",
    id_vars = oak_id_vars()
  )
```

As you can see in this function call, the raw dataset and variable names are passed as arguments. As long as the raw dataset and variable are present in the global environment, the function will execute the algorithm's logic and create the target SDTM variable.

{sdtm.oak} is designed to handle any type of raw input format. It is not tied to any specific data collection standards, making it both EDC-agnostic and data standards-agnostic.

PROGRAMMING WORKFLOW

The programming steps concept is close to the key SDTM concepts, that is to map the topic variables first, and then map its qualifiers and identifiers. The programming steps are generic across multiple SDTM domain classes like events, interventions and findings.

In {sdtm.oak} we process one raw dataset at a time. Similar raw datasets can be stacked together before processing. Below are the typical programming steps for an SDTM domain:

- Read in data
- Create oak_id_vars
- Read in controlled terminologies
- Map topic variable
- Map rest of the variables
- Repeat map topic and map rest

Repeat the above steps for different raw datasets before proceeding with the below steps.

- Create SDTM derived variables
- Add labels and attributes

ROADMAP

We are planning to develop the below features in the subsequent releases:

- Functions required to derive reference date variables in the DM domain.
- Metadata driven automation based on the standardized SDTM specification.
- Functions required to program the EPOCH variable.
- Functions to derive standard units and results based on metadata.
- Functions required to create SUPP domains.
- Making the algorithms part of the standard CDISC eCRF portal enabling automation of CDISC standard eCRFs.

REFERENCES

Ganapathy, Rammprasad. 2024. "Introducing Sdtm.oak." October 24, 2024. https://pharmaverse.github.io/blog/posts/2024-10-24_introducing.../introducing_sdtm.oak.html.

ACKNOWLEDGMENTS

We thank the contributors and authors of the package. We also thank the CDISC COSA for sponsoring the {sdtm.oak}. Additionally, we would like to sincerely thank the volunteers from Roche, Pfizer, GSK, Vertex, and Merck for their valuable input as integral members of the CDISC COSA - OAK leadership team.

RECOMMENDED READING

https://pharmaverse.github.io/sdtm.oak/index.html

https://pharmaverse.github.io/blog/posts/2024-10-24_introducing.../introducing_sdtm.oak.html

CONTACT INFORMATION

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

Author Name: Rammprasad Ganapathy
Company: Roche/Genentech
Work Phone: +1 617-538-8745
Email: ganapar1@gene.com
Website: https://www.gene.com

Author Name: Shiyu Chen
Company: Atorus Research
Work Phone: +1 267-665-9773
Email: Shiyu.Chen@atorusresearch.com
Website: https://www.atorusresearch.com

The OAK team can be reached through any of the following means:
Slack: https://oakgarden.slack.com,
GitHub: https://github.com/pharmaverse/sdtm.oak,
CDISC Wiki: https://wiki.cdisc.org/display/oakgarden,