PharmaSUG China 2024 - Paper AA-10007

Auto-annotation CRF per SDTM-MSG 2.0 by R

Qi Luo, SinoCellTech

ABSTRACT

Preparing the aCRF is a tedious and time-outing work for clinical programmer and it's an essential component part of eSub. We will discuss how to complete the SDTM annotation from unique blank CRF by R. And more details will be presented to meet the regulation and streamline the process of SDTM annotation per SDTM-MSG 2.0 in this paper.

INTRODUCTION

Generally, it's convenient to generate aCRF if we have a referenced aCRF of similar clinical trials. We can copy or auto-annotate a lot of annotations by manual or practical tools. But how can we complete auto-annotations efficiently if we don't have reference aCRF. We will discuss the process from unique blank CRF to aCRF step by step in this paper.

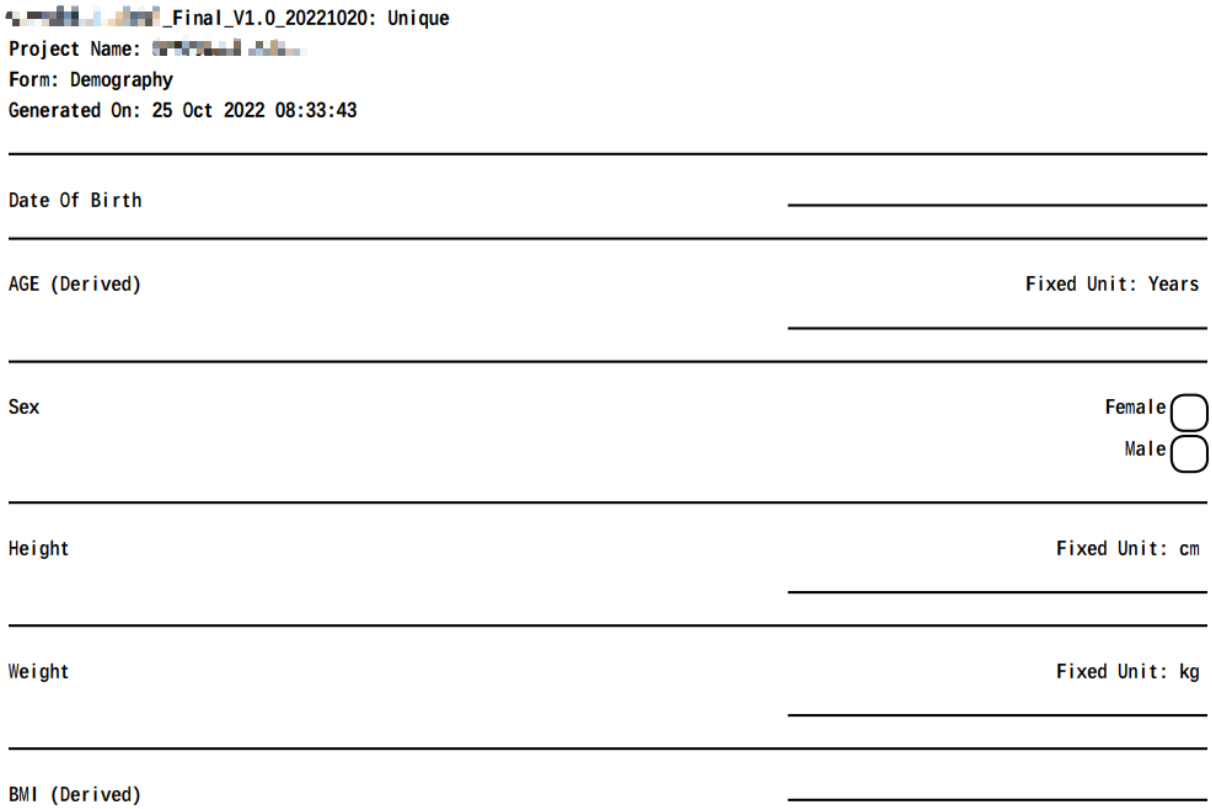
OVERVIEW OF METHODOLOGY

1. We need to know the coordinates, height and weight of the text from CRF if we want to place annotations in the appropriate position of CRF. "pdftools" package from R can help us extract metadata information of CRF.
2. We can create a template of annotation to add annotations conveniently, distinguish different annotation text and define annotations rules.
3. Add annotations to template according to our understanding of domain and defined rules.
4. Transfer the annotations to XFDF and import it to PDF files.

METHODOLOGY

STEP 1: EXTRACT METADATA FROM CRF.PDF

It's important to get the coordinate information of text if we want to add annotation to the proper position of aCRF.pdf. For example, it's necessary to get the coordinate, height, width of the text box "Date of Birth" from CRF as below, because they decided where to add annotation "BRTHDTC".



Display 1. Demography Page of Unique Blank CRF

The following R packages should be loaded in advance, “pdftools” package can extract text, fonts, attachments and metadata from a pdf file and “rlist” packages can manipulate list elements conveniently:

```
library(pdftools)
library(rlist)
```

“file.path” function is used to load the unique blank CRF.pdf, “pdf_data” and “list.stack” function are used to extract the text box data and stack all list elements to tabular data and then output the metadata:

```
pdf_file <- file.path("unique.pdf")
data <- pdf_data(pdf_file)
df_data <- list.stack(data, data.table = TRUE)
write.csv(df_data, file = "df_data.csv")
```

From the screenshot of “df_data.csv”, we have got all useful information of text box.

Display 2. Screenshot of df_data.csv

STEP 2: GENERATE ANNOTATION TEMPLATE

Connect the text that should be continues through the logical variable “space” and update the x, y, width, height of connected text box. And then we defined 3 header variables and 10 annotation variables to add annotation later. We can also add more variables if necessary:

```
df_data1 <- df_data %>%
  mutate(page = 0,
         tf = x <= lag(x) & y < lag(y),
         tf1 = ifelse(is.na(tf), FALSE, tf),
         page = cumsum(tf1 + 0L)) %>%
  select(!c(tf, tf1))

df_data2 <- df_data1 %>%
  mutate(
    spacegroup = cumsum(ifelse(space == TRUE, FALSE, TRUE) + 0L),
    spacegroup = ifelse(space == FALSE, spacegroup-1, spacegroup)) %>%
  group_by(spacegroup) %>%
  mutate(Ttext = paste0(text, collapse = " "),
         Tx = nth(x, 1),
        Twidth = x-Tx+width) %>%
  ungroup() %>%
  filter(space == FALSE, 110 < y, y < 700) %>%
  mutate(rownum = row_number()) %>%
  select(rownum, page, Tx, y, Twidth, height, Ttext) %>%
  rename(x = Tx, width = Twidth, text = Ttext)

df_data2[, paste0("header", 1:3)] = ""
df_data2[, paste0("annotation", 1:10)] = ""
```

```
writexl::write_xlsx(df_data2, path = "annotation.xlsx")
```

STEP 3: ADD ANNOTATION TO ANNOTATION.XLSX

This is an example of annotation.xlsx and we have added annotations in column “header1-headerX” and column “annotation1-annotationX” as need. And annotations can consist of up to three parts.

A	B	C	D	E	F	G	H	I	J	K	L
rownum	page	x	y	width	height	text	header1	header2	header3	annotation1	annotation2
3	1	72	143	84	8	Informed Consent Date				DM.RFICDTC	DS.DSSTDTC
4	2	72	143	52	8	Date Of Birth				DM.BRTHDTC	
5	2	72	176	52	8	AGE (Derived)				DM.AGE	
6	2	72	224	12	8	Sex				DM.SEX	
7	2	469	176	68	8	Fixed Unit: Years				DM.ASSIGNED.AGEU	
10	2	72	279	24	8	Height				VS.ASSIGNED.VSTESTCD = HEIGHT	VS.VSORRES / VSORRESU when VSTESTCD = HEIGHT
11	2	481	279	56	8	Fixed Unit: cm					
12	2	72	327	24	8	Weight				VS.ASSIGNED.VSTESTCD = WEIGHT	VS.VSORRES / VSORRESU when VSTESTCD = WEIGHT
13	2	481	327	56	8	Fixed Unit: kg					
14	2	72	375	52	8	BMI (Derived)				VS.ASSIGNED.VSTESTCD = BMI	VS.VSORRES when VSTESTCD = BMI
15	2	72	408	36	8	Ethnicity				DM.ETHNIC	
20	2	72	495	16	8	Race				DM.RACE	
37	4	72	143	136	8	Were Any Medical History Reported?	MH.ASSIGNED.MHCAT = Medical History			MH.ASSIGNED.[NOT SUBMITTED]	
40	4	72	198	100	8	Disease/Surgery/Allergies				MH.MHTERM	
41	4	72	231	40	8	Start Date				MH.MHSTDTC	

Display 3. Screenshot of annotation.xlsx

How to Select Column “headerX” and “annotationX”

- Type text in column “annotation1” if there is only one annotation for the question.
- Type text in column “annotation2-annotationX” if there are more annotations for the same question.
- Type text in column “headerX” if we want to explain a situation on the CRF and not direct collected-variable annotations.

How to Type Annotation

- The first part of separated by periods refer to which domain the annotation belongs to.
- The second part of separated by periods refer to whether the annotation is assigned and not direct collected from variable.
- The third part of separated by periods refer to the annotation label we want to demonstrate on the CRF.

STEP 4: GENERATE ACRF.XFDF

Firstly, prepare the sdtm_header.xlsx (refer to the display as below) to demonstrate domain description later and import completed annotation.xlsx as “anno1” object and sdtm_header.xlsx as “sdtm_header” object.

A	C
Domain	**Label_en**
CO	Comments
DM	Demographics
SE	Subject Elements
SV	Subject Visits
CM	Concomitant/Prior Medications
EC	Exposure as Collected
EX	Exposure
PR	Procedures
SU	Substance Use
AE	Adverse Events
CE	Clinical Events
DS	Disposition
DV	Protocol Deviations
HO	Healthcare Encounters
MH	Medical History
DA	Drug Accountability
DD	Death Details
EG	ECG Test Results

Display 4. Screenshot of sdtm_header.xlsx

Generate the Annotation Label Displayed on the CRF

Transpose the column “annotationX” by “pivot_longer” function and split it to “domain”, ”assigned” and “annotext” column by regular expression:

```
anno2 <- anno1 %>%
  pivot_longer(cols = paste0("annotation", 1:10),
               names_to = "annord_label",
               values_to = "anno") %>%
  filter(!is.na(anno)) %>%
  mutate(annord = as.numeric(str_extract_all(annord_label, "[0-9]")),
         domain = str_split_fixed(anno, "\\.", n = 3)[,1],
         assigned = str_split_fixed(anno, "\\.", n = 3)[,2],
         annotext = str_split_fixed(anno, "\\.", n = 3)[,3]) %>%
  mutate(annotext = ifelse(assigned == "ASSIGNED", annotext, assigned),
         assigned = ifelse(assigned == "ASSIGNED", assigned, NA),
         cat = 2)
```

Then perform the similar data process as above for column “headerX” and combine the two dataframe from “headerX” and “annotationX”.

```
anno3 <- left_join(anno2, sdtm_header[c("Domain", "Label_en")], by =
c("domain" = "Domain")) %>%
  mutate(header0 = paste0(domain, ".", domain, "(", Label_en, ")")) %>%
  pivot_longer(cols = paste0("header", 0:3),
               names_to = "headord_label",
               values_to = "hanno") %>%
  filter(!is.na(hanno)) %>%
  mutate(headord = as.numeric(str_extract_all(headord_label, "[0-
9]"))+1) %>%
  distinct(page, hanno, .keep_all = TRUE) %>%
  select(!c(domain, annotext, assigned)) %>%
  mutate(domain = str_split_fixed(hanno, "\\.", n = 3)[,1],
         assigned = str_split_fixed(hanno, "\\.", n = 3)[,2],
         annotext = str_split_fixed(hanno, "\\.", n = 3)[,3]) %>%
  mutate(annotext = ifelse(assigned == "ASSIGNED", annotext, assigned),
         assigned = ifelse(assigned == "ASSIGNED", assigned, NA),
         cat = 1)
```

```
annoall <- rbind.fill(anno2, anno3)
```

Derive Necessary Variables for aCRF.xfdf

Table 1 is a necessary variable list for aCRF.xfdf.

Variable	Description
pagec	CRF.pdf page number
backcolor	Color of background
coord	Coordinate of text box
annotext	Value of text box

Table 1. Variable List

Page number in .xfdf:

```
pagec = as.character(page),
```

Set different background color for different domain on a single CRF page:

```
backcolor = case_when(colorord == 1 ~ "#BFFFFF",
                  colorord == 2 ~ "#FFFFAA",
                  colorord == 3 ~ "#96FF96",
                  colorord == 4 ~ "#FFBE96",
                  colorord == 5 ~ "#1CBBEB",
                  colorord == 6 ~ "#CA7EEF"),
```

Set consistent font size:

```
fontsize = 12,
```

Calculate the height and width of text box:

```
boxheight = ifelse(cat == 1, 18, 15),

adlength = (nchar(str_replace_all(annotext, " ", ""), type = "bytes")-
nchar(str_replace_all(annotext, " ",
"")))/3+nchar(str_replace_all(annotext, " ", "")),

boxlength = if_else(headord == 1 & !is.na(headord), fontsize*0.7*adlength,
fontsize*0.7*adlength+12)
```

Derive variable x2, y2, coord:

```
x2 = x1 + boxlength,
y2 = y1 + boxheight,
coord = paste(x1,y1,x2,y2,sep=",")
```

Output aCRF.xfdf

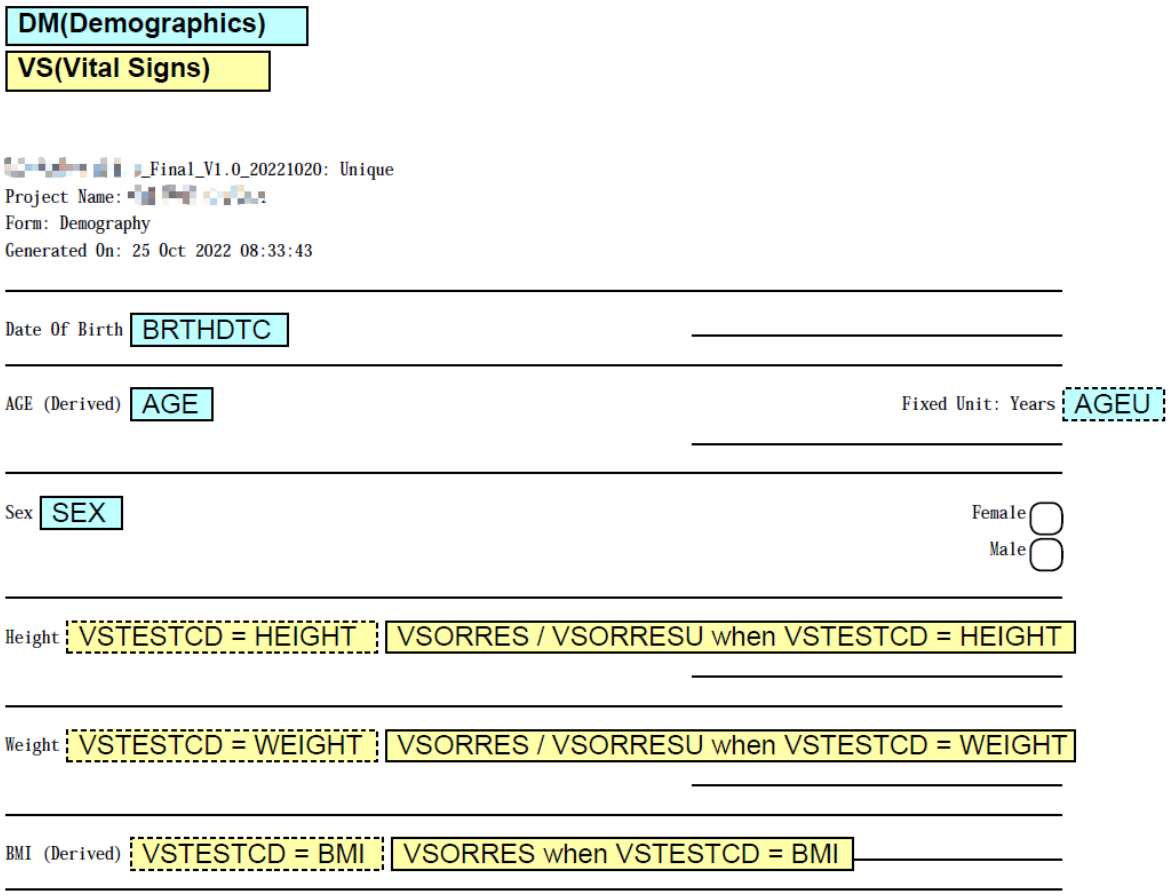
A file with .xfdf extension is an XML Forms Data Formats that is generated with Adobe Acrobat software. XFDF files are saved in XML file format that is a universal format used for import and export of data. “xml2” package in R helps us manage and ouput .xfdf files very well.

Pay attention to the “ASSIGNED” flag and we have to draw a dashed text box per SDTM-MSG 2.0. And then set the bold format for domain description.

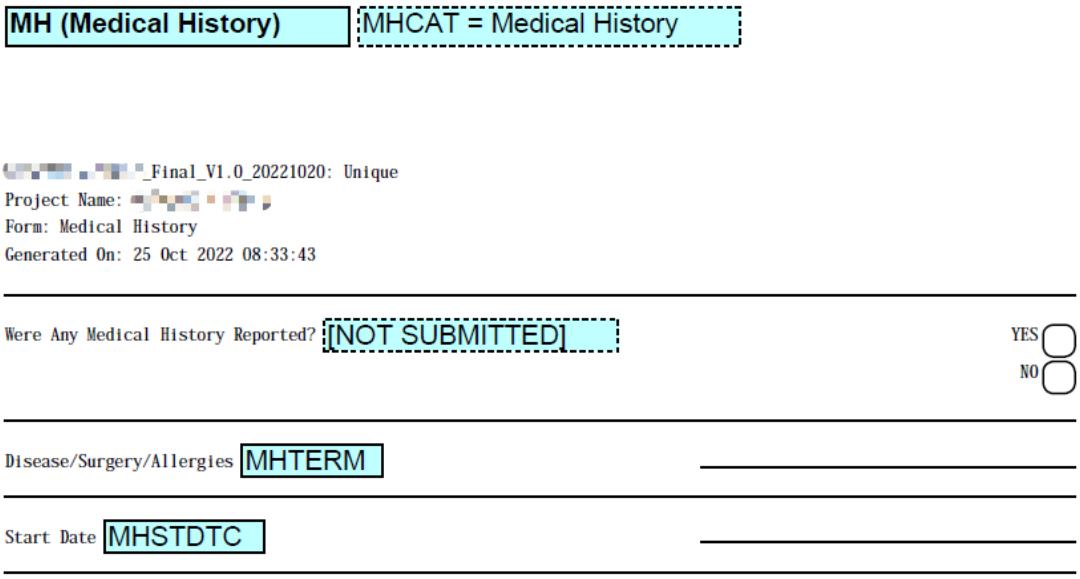
```
library(xml2)
bign <- as.numeric(count(final))
xmlanno <- paste("<?xml version=\"1.0\" encoding=\"UTF-8\"?>",
                 "<xfdf xmlns=\"http://ns.adobe.com/xfdf/\"
xml:space=\"preserve\">",
```

```
                "<annots>")

for (i in 1:bign) {
  xmlanno <- paste(xmlanno,
                   paste0(ifelse(final["assigned"][i,] == "ASSIGNED"
& !is.na(final["assigned"][i,]),
                                  paste0("<square width=\"1\" dashes=\"2,2\"
style=\"dash\" color=\"#000000\" page=\"",
                                         final["pagec"][i,],
                                         "\" rect=\"",
                                         final["coord"][i,],
                                         "\" interior-color=\"",
                                         final["backcolor"][i,],
                                         "\"><popup
flags=\"print,nozoom,norotate\" open=\"no\" rect=\"",
                                         final["coord"][i,],
                                         "\"/></square>"),
                                  "")),
                   paste0("<freetext page =\"",
                          final["pagec"][i,],
                          "\" rect=\"",
                          final["coord"][i,],
                          "\" subject=\"",
                          final["domain"][i,],
                          paste0(ifelse(final["assigned"][i,] == "ASSIGNED"
& !is.na(final["assigned"][i,]),
                                         # "","
                                         paste0("\" width=\"0"),
                                         paste0("\" color=\"",
                                                final["backcolor"][i,])
                          )),
                          "\">"),
                   paste0("<contents-richtext>"),
                   paste0("<body xmlns=\"http://www.w3.org/1999/xhtml\"
xmlns:xfa=\"http://www.xfa.org/schema/xfa-data/1.0/\"
xfa:APIVersion=\"Acrobat:11.0.0\" xfa:spec=\"2.0.2\"
                   style=\"text-align:left;
                   font-weight:bold;
                   font-family: Arial;
                   font-stretch:normal;
                   font-style:normal;
                   font-size:12 pt;
                   color:#000000;
                   font-weight:",final["fontweight"][i,],";
                   \" >"),
                   paste0("<p dir=\"ltr\">",final["annotext"][i,],"</p>"),
                   paste0("</body>","</contents-richtext>","</freetext>")
  )
}

xmlanno <- paste(xmlanno, "</annots></xfdf>")
write_xml(read_xml(xmlanno), "acrf.xfdf", options = "format", encoding =
"UTF-8")
```

Screenshot Presentation



Display 5. Screenshot 1 of presentation



Display 6. Screenshot 2 of presentation

CONCLUSION

This paper presented a method to generate XFDF efficiently and we can update the annotation text or deal with the situation that CRF was updated easily through this method. And more and more clinical programmers have learned R language, this method can serve as a reference and we will create more powerful tools.

REFERENCES

Noory Kim. 2021. “Annotating CRFs More Efficiently.” PharmaSUG 2021 - Paper AD-036

CONTACT INFORMATION

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

Qi Luo
Clinical Programming
Sinocelltech Ltd.
qi_luo@sinocelltech.com