Implementing FDA Specifications in R for Submission of Analysis Datasets

Timothy Bergsma*[1], Scott Pivirotto[2]
[1]Certara Consulting Services, Cary, NC, USA
[2]Metrum Research Group, LLC, Tariffville, CT, USA

Objectives

Pharmacometric analysis datasets are subject to considerable formatting requirements for submission to the FDA [1,2]. Our objective was to create a utility in the R programming language [3] to generate an integrated deliverable conforming to FDA guidance.

Methods

We developed a function `define` [4] that processes a list of file paths. Files are copied to a user-identified output directory or subdirectories. Files in CSV format are converted to SAS TRANSPORT format by means of the R package `SASxport` [5]. Text files are copied verbatim, appending the TXT extension as necessary.

The file `define.pdf`, created in the top directory, includes a table of contents with external hyperlinks to the archived files and internal hyperlinks to data definition tables. Column definitions are read from metadata files as per `specification` in `metrumrg` version 5.57 [6]. Rendering requires LaTeXand `metrumrg` utilities.

Results

The `define` utility was tested in a production envrironment and by means of a reproducible example (below). Resulting directory structure and `define.pdf` (Figure 1, Figure 2) were satisfactory. Names and labels were automatically coordinated among transport files, tables of contents, and definition tables. The process was fast and accurate, with each discretionary value (e.g. dataset name) supplied only once.

Dependencies

In addition to installing R, it is necessary to have LaTeXconfigured. See the excellent advice at http://tinyurl.com/rstudio-latex.

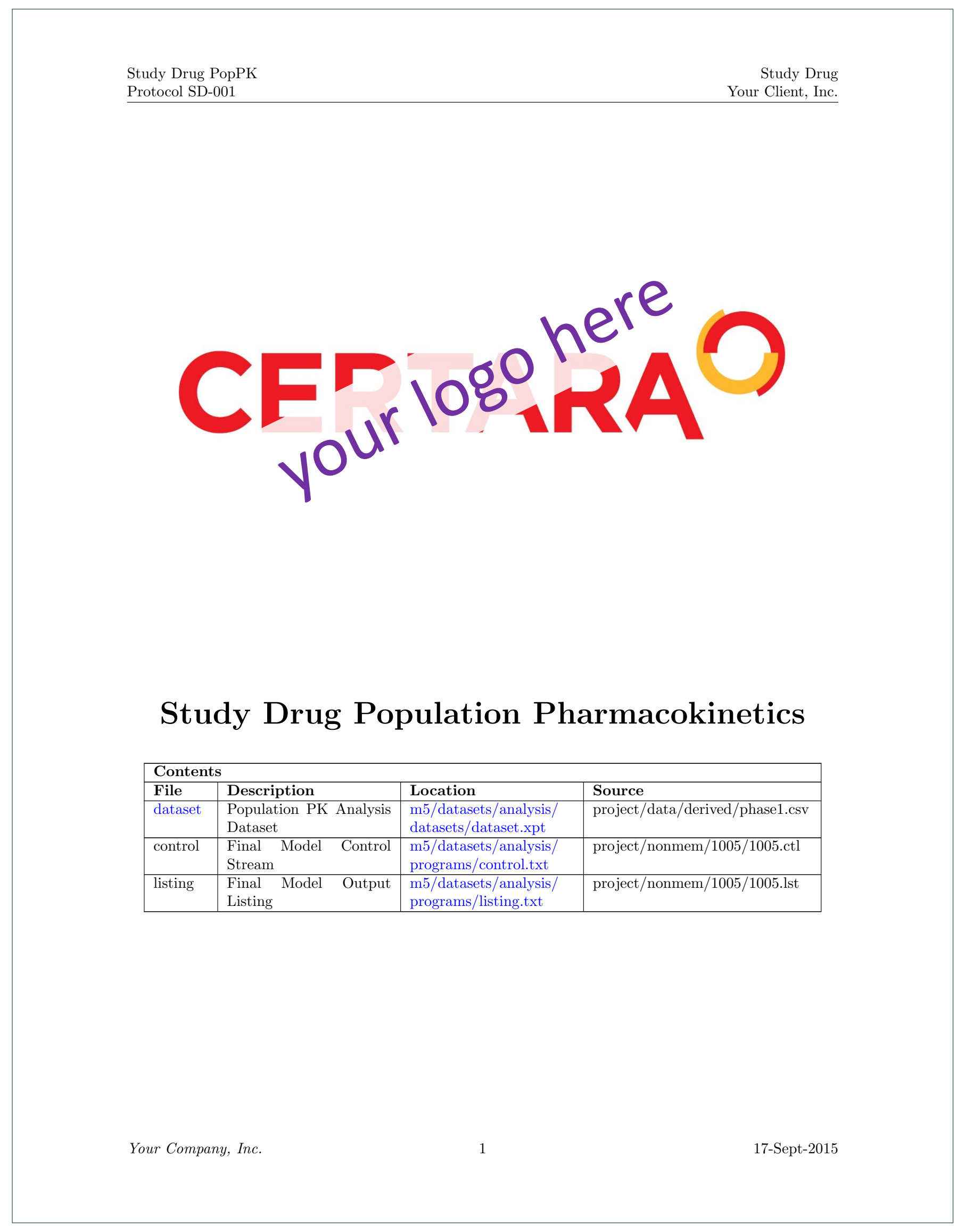
The R packages `metrumrg`, `define`, and their dependencies (especially `SASxport`) should be installed.

```
dir.create( 'lib' )
.libPaths( 'lib' )
install.packages(
  c( 'metrumrg', 'SASxport', 'devtools' ),
  repos=c(
    'http://R-Forge.R-project.org',
    'http://cran.rstudio.com' ))
library( devtools )
install_github(
  'bergsmat/define',
  ref='f5e6a26b1a00c9c57f277808c29bb9008eb5408b'
)
  # ref identifies the version used here.
  # Later versions may be available.
```

Environment

For simplicity, we make a local copy of the example project.

```
file.copy(
  from = 'lib/metrumrg/example/project',
  to = '.',recursive = TRUE
)
library( define )
```



Study Drug PopPK
Protocol SD-001

Study Drug
Your Client, Inc.

Study Drug Population Pharmacokinetics

Contents			
File	**Description**	**Location**	**Source**
dataset	Population PK Analysis Dataset	m5/datasets/analysis/datasets/dataset.xpt	project/data/derived/phase1.csv
control	Final Model Control Stream	m5/datasets/analysis/programs/control.txt	project/nonmem/1005/1005.ctl
listing	Final Model Output Listing	m5/datasets/analysis/programs/listing.txt	project/nonmem/1005/1005.lst

Your Company, Inc. 1 17-Sept-2015

Figure 1: Page 1 of define.dpf

Metadata

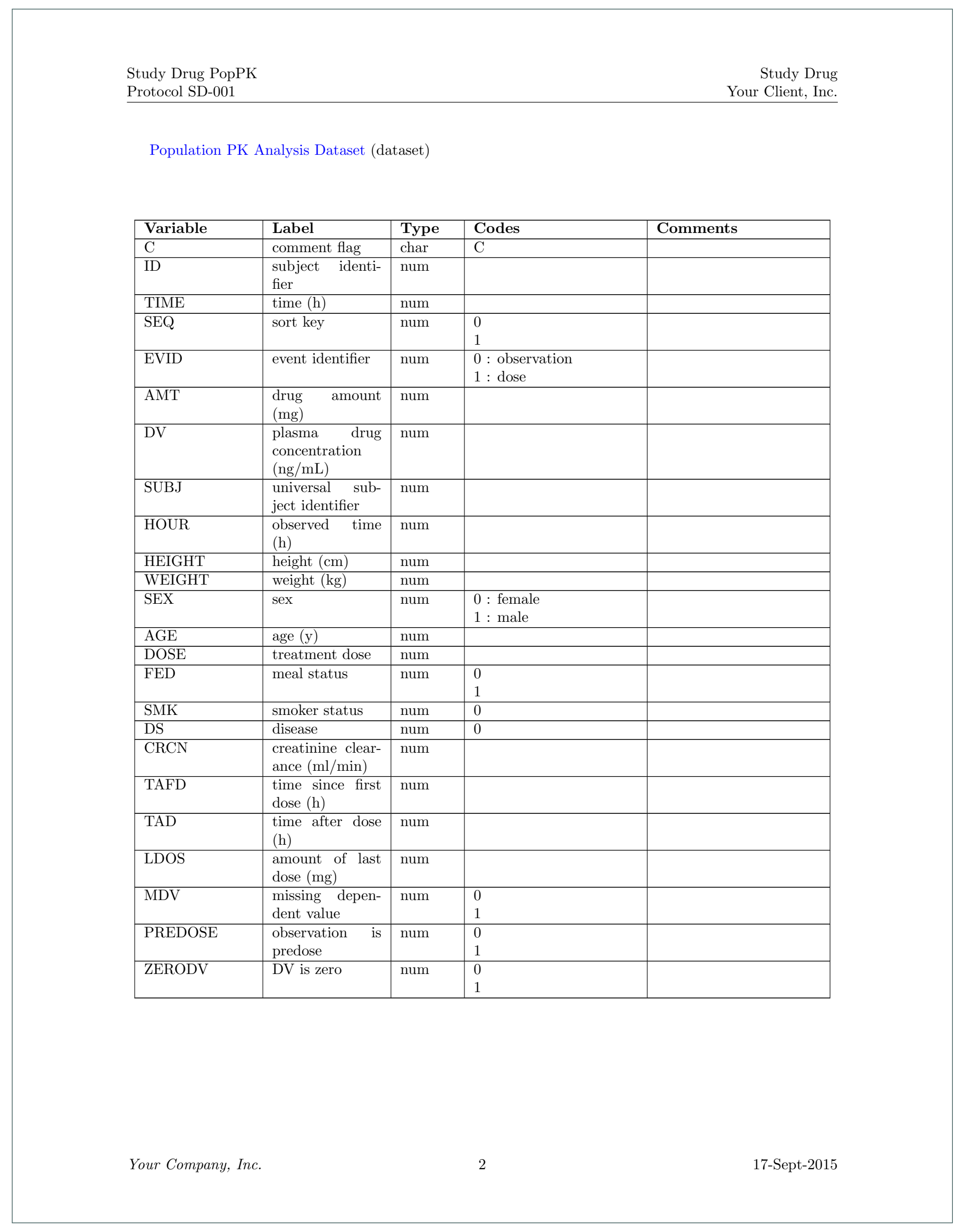
We create metadata. In particular, we are interested in providing definitions for the columns in the analysis dataset. These are used to populate fields in the SAS TRANSPORT file, as well as `define.pdf`.

```
m <- specification(
  read.csv(
    'project/data/derived/phase1.csv',
    na.strings='.' ))

m$label <- c(
  'comment flag','subject identifier','time (h)',
  'sort key','event identifier','drug amount (mg)',
  'plasma drug concentration (ng/mL)',
  'universal subject identifier','observed time (h)',
  'height (cm)','weight (kg)','sex','age (y)',
  'treatment dose','meal status','smoker status',
  'disease','creatinine clearance (ml/min)',
  'time since first dose (h)','time after dose (h)',
  'amount of last dose (mg)','missing dependent value',
  'observation is predose','DV is zero')
```

While the `specification` strategy in `metrumrg` allows units of continuous variables to be defined in the `guide` column, we have defined them directly in the label for convenience. ‘Decodes’ for factor-like variables are still defined in the guide column. We give examples for EVID and SEX. This information will find its way to `define.pdf` (Figure 2).

```
m$guide[m$column=='EVID'] <-
  encode(0:1, labels=c('observation','dose'))
m$guide[m$column=='SEX'] <-
  encode(0:1, labels=c('female','male'))
m$guide[m$column %in% c('AMT','LDOS','DOSE')] <-
  'mg [1000,100000]' # replace default encoding
```



Study Drug PopPK
Protocol SD-001

Study Drug
Your Client, Inc.

Population PK Analysis Dataset (dataset)

Variable	Label	Type	Codes	Comments
C	comment flag	char	C	
ID	subject identifier	num		
TIME	time (h)	num		
SEQ	sort key	num	0 1	
EVID	event identifier	num	0 : observation 1 : dose	
AMT	drug amount (mg)	num		
DV	plasma drug concentration (ng/mL)	num		
SUBJ	universal subject identifier	num		
HOUR	observed time (h)	num		
HEIGHT	height (cm)	num		
WEIGHT	weight (kg)	num		
SEX	sex	num	0 : female 1 : male	
AGE	age (y)	num		
DOSE	treatment dose	num		
FED	meal status	num	0 1	
SMK	smoker status	num	0	
DS	disease	num	0	
CRCN	creatinine clearance (ml/min)	num		
TAFD	time since first dose (h)	num		
TAD	time after dose (h)	num		
LDOS	amount of last dose (mg)	num		
MDV	missing dependent value	num	0 1	
PREDOSE	observation is predose	num	0 1	
ZERODV	DV is zero	num	0 1	

Your Company, Inc. 2 17-Sept-2015

Figure 2: Page 2 of define.pdf

Metadata is saved as a tab-delimited companion of the data file. It can be further edited with an external utility, taking care to preserve tab formatting.

```
write.spec(m,'project/data/derived/phase1.spec')
```

Invocation

Finally, we call the function `define`, passing a vector of paths to files that will be documented. Here we document the analysis dataset, as well as the control stream and output for a related NONMEM run.

```
define(
  x=c(
    dataset='project/data/derived/phase1.csv',
    control='project/nonmem/1005/1005.ctl',
    listing='project/nonmem/1005/1005.lst'
  ),
  subdir=c(
    'm5/datasets/analysis/datasets',
    'm5/datasets/analysis/programs',
    'm5/datasets/analysis/programs'
  ),
  description=c(
    'Population PK Analysis Dataset',
    'Final Model Control Stream',
    'Final Model Output Listing'
  ),
  title = 'Study Drug Population Pharmacokinetics',
  short = 'Study Drug PopPK',
  protocol = 'Protocol SD-001',
  sponsor = 'Your Client, Inc.',
  program = 'Study Drug',
  author = 'Your Company, Inc.',
  logo = 'yourLogoHere.pdf'
)
```

At this point, a directory tree has been created, with archived files and a pdf document describing them.

```
dir('define',full.names=TRUE, recursive=TRUE)
```

```
[1] "define/define.pdf"
[2] "define/m5/datasets/analysis/datasets/dataset.xpt"
[3] "define/m5/datasets/analysis/programs/control.txt"
[4] "define/m5/datasets/analysis/programs/listing.txt"
```

Discussion

Functionalized generation of define documentation reduces effort and minimizes errors. Many details must be tightly coordinated, such as the placement of files and the hyperlinks that point to them, or the column labels in the transport file and the column labels in the definition table. Certain transformations are deterministic, such as the routine replacement of file extensions, and therefore beg for automation. Manual approaches can be tedious and may lead to errors, e.g. where required substitutions of template values are overlooked.

The `define` approach relies systematically on singular specification of each semantic element. For example, the definition of each data column is supplied once in a metadata file, and re-used programatically at multiple points in the output. Similarly, aesthetic features – such as document names and header values – are supplied once during the function call, and re-used as necessary.

Conclusions

Use of the `define` function allowed rapid, accurate preparation of a submission-ready directory containing archived pharmacometric analysis datasets and coordinated documentation.

References

[1] U.S. Food and Drug Administration (2012). *Study Data Specifications. Version 2.0.* http://tinyurl.com/nqcp7u3

[2] U.S. Food and Drug Administration (2014). *Portable Document Format (PDF) Specifications.* http://tinyurl.com/onocmc5

[3] R Core Team (2014). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org

[4] https://github.com/bergsmat/define

[5] Warnes G. *SASxport: Read and Write SAS XPORT Files (2014).* R package version 1.5.0. http://cran.r-project.org/src/contrib/Archive/SASxport/

[6] Bergsma TT, Knebel W, Fisher J, Gillespie WR, Riggs MM, Gibiansky L and Gastonguay MR (2013). *Facilitating pharmacometric workflow with the metrumrg package for R.* Computer Methods and Programs in Biomedicine 109 (1): 77-85. http://www.cmpbjournal.com/article/S0169-2607(12)00191-5/abstract