File Reformatting Example

James Eddy

January 26, 2015

First, load the required libraries.

library(synapseClient)  
library(xlsx)  
library(dplyr)  
library(knitr)

Login to Synapse using credentials saved in .synapseConfig file.

synapseLogin()

## Welcome jaeddy!

Using functions in synapseClient and xlsx, load data from the Synapse object into R.

# Define paths for required Synapse objects  
unformatted\_gwas\_covars\_address <- "syn3025476" # covariates Excel file  
  
# Download files from Synapse  
covars\_file <- synGet(unformatted\_gwas\_covars\_address)  
covars\_file\_path <- getFileLocation(covars\_file)  
  
# Load the files into R  
unformatted\_gwas\_covars <- read.xlsx2(covars\_file\_path, 2,   
 stringsAsFactors = FALSE)

AMP-AD partner teams were provided with a template to follow for providing and formatting clinical variables. I've used the column headers from this template to display an example, but I'll also use them to rearrange the Mayo data below.

# Define column headers from template  
headers <- c("participant\_id", "age\_at\_onset", "age\_at\_last\_assessment",   
 "age\_at\_death", "post\_mortem\_interval", "sex", "education",  
 "apoe\_genotype", "race\_ethnicity", "braak\_stage", "mmse\_at\_onset",  
 "mmse\_at\_last\_assessment", "cerad")  
  
template <- data\_frame("sample1", 75, 86, 86, 10, "M", 12, "E3E4",   
 "European American", "V", 24, 18, 30)  
names(template) <- headers  
template[2, ] <- c("sample100", 82, 88, 89, 4, "F", 17, "E2E2",   
 "African American", "VI", 23, 17, 17)

Here's what the template looks like:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| participant\_id | age\_at\_onset | age\_at\_last\_assessment | age\_at\_death | post\_mortem\_interval | sex | education | apoe\_genotype | race\_ethnicity | braak\_stage | mmse\_at\_onset | mmse\_at\_last\_assessment | cerad |
| sample1 | 75 | 86 | 86 | 10 | M | 12 | E3E4 | European American | V | 24 | 18 | 30 |
| sample100 | 82 | 88 | 89 | 4 | F | 17 | E2E2 | African American | VI | 23 | 17 | 17 |

For comparison, here is what the Mayo table looks like, before reformatting:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FID | IID | RSGWAS | AUTGWAS | AgeOver60 | Sex | APOE4\_Dose |
| 0 | 1820676577\_B | 1 | 0 | 10.74 | 0 | -9 |
| 0 | 1823480380\_B | 1 | 0 | 12.18 | 0 | -9 |
| 0 | 1806081232\_A | 1 | 0 | 12.84 | 0 | -9 |
| 0 | 1827270180\_B | 1 | 0 | 13.38 | 0 | -9 |
| 0 | 1823480567\_A | 1 | 0 | 15.14 | 0 | -9 |
| 0 | 1823480378\_B | 1 | 0 | 16.13 | 0 | -9 |

Now for the fun part! I'll rearrange and reformat the data to fit the template, placing NA entries for any information not included in the Mayo table.

# Simple function to recode APOE genotype value  
recode\_apoe\_status <- Vectorize(function(apoe\_status) {  
 if (apoe\_status == 1) {  
 "E4"  
 } else if (apoe\_status == 0) {  
 "none"  
 } else {  
 NA  
 }  
})  
  
# Do the reformatting with dplyr (and pipes!)  
gwas\_covars <- unformatted\_gwas\_covars %>%   
 # pull out relevant variables from original data frame  
 select(participant\_id = IID, age\_at\_diagnosis = AgeOver60,   
 sex = Sex, apoe\_genotype = APOE4\_Dose) %>%  
   
 # rename and modify variables to match template  
 mutate(age\_at\_last\_assessment = as.numeric(age\_at\_diagnosis) + 60,  
 sex = ifelse(sex == 1, "M", "F"),  
 apoe\_genotype = recode\_apoe\_status(apoe\_genotype)) %>%  
   
 # add empty columns for additional template variables  
 mutate(age\_at\_onset = NA,  
 age\_at\_death = NA,  
 post\_mortem\_interval = NA,  
 education = NA,  
 race\_ethnicity = NA,  
 braak\_stage = NA,  
 mmse\_at\_onset = NA,  
 mmse\_at\_last\_assessment = NA,  
 cerad = NA) %>%  
   
 # reorder variables to match template column order  
 select(one\_of(headers))

The following code is used to upload the new, reformatted data back to our Synapse site.

# Write new data frame to text file  
file\_path <- file.path(tempdir(), "mayo\_gwas\_clinical\_vars.txt")  
write.table(gwas\_covars, file\_path)  
  
formatted\_covars\_file <- File(path = file\_path,   
 parentId = covars\_file$properties$parentId)  
formatted\_covars\_file <- synStore(formatted\_covars\_file)

Importantly, I can use my custom build\_relationship function to define the provenance for this activity, linking input to code to output. Note: I'll need to check on Synapse for the object ID of the file I just uploaded.

source("synapseHelpers/R/set\_provenance.R")  
  
activity\_name <- "clinical data formatting"  
input\_files <- c("syn3025476")  
  
code\_address <- paste0("https://github.com/jaeddy/ampSynapseProjects/",  
"raw/master/synapseUploads/R/format\_gwas\_covars.R")  
code\_files <- list(list(name = "format\_gwas\_covars.R",  
 url = code\_address))  
output\_files <- c("syn3026432")  
description <- paste("To execute run:",   
 "Rscript format\_gwas\_covars.R")  
  
build\_relationship(activity\_name, input\_files, code\_files,   
 output\_files, description)