ADNI Analysis: Data Management

Justin M. Leach

1 Introduction

In other projects we included data management and data analysis in the same file, but this is too cluttered. We use the same ADNI data for several projects and this file discusses topics such as:

(1) How the data were cleaned/processed after obtaining from ADNI.
(2) Generating neighborhood matrices for when spatial priors are included.

Note that I often comment out `saveRDS()`, but if you're running things from scratch you'll want to uncomment these and save the files in whatever directory you desire. Note also that the `R` code used in this paper employed `R` version 3.6.3 for data management and tables/figures.[1] As discussed below, analyses were run on UAB's Cheaha, where `R` version 3.6.0 was employed. If you run into issues with re-running this code, then you may need to check that you are using the same version of `R`. The `R` package `tidyverse` is version 1.3.0.

2 Cleaning/Processing Data

The `ADNIMERGE` package has all the relevant data. We use data that has been processed with cross sectioal FreeSurfer (6.0). The data will consist of regional summaries on the Desikan-Killiany Atlas. Note that the ADNIMERGE package is often updated with new data, and we used the version packaged on 2020-02-19.

```
library(tidyverse)
library(ADNIMERGE)
# help(package = "ADNIMERGE")
# names(adnimerge)
# data("datadic")
data("ucsffsx6") # Cross sectional FreeSurfer (6.0)
```

There are more variables than we need, so we specify which ones to keep. On order, we keep diagnosis (DIAGNOSIS), subject identifiers (RID), visit identifier (VISCODE), and the date of assessment (EXAMDATE), and whether the images passed quality control measures (OVERALLQC); we also include several demographic variables. Note that the exam dates may not be the same for diagnosis and imaging, as these assessments may not occur on exactly the same day.

```
keep.dx <- c("DIAGNOSIS")
keep.all <- c("RID", "VISCODE", "EXAMDATE")
keep.imd <- c("OVERALLQC")
keep.demo <- c("AGE", "PTGENDER", "PTEDUCAT", "PTETHCAT", "PTRACCAT")
```

[1] Initial analyses were conducted in 'R' version 3.6.3, but we experienced no issues with compilation after updating to 'R' version 4.1.0.

```
dx.redux <- dxsum[, c(keep.all, keep.dx)]
mri.info <- ucsffsx6[, c(keep.all, keep.imd)]
demo <- adnimerge[, c(keep.all, keep.demo)] %>%
  mutate(RID = as.numeric(RID),
         VISCODE = as.character(VISCODE))
```

2.1 Cortical Thickness

We only retain subjects who fully passed quality control, remove any subjects who have MRI but lack a diagnosis, removed several duplicate entries, and used screening visit as the cross section.

```
attributes(mri.info$VISCODE) <- NULL

# Just keep subjects who had full passes
check.out <- merge(x = dx.redux, y = mri.info, by = c("RID", "VISCODE"))
check.out <- check.out[order(check.out$RID), ]
pass.out <- check.out[check.out$OVERALLQC == "Pass", ]
# table(pass.out$VISCODE)

# Which have mri but not diagnosis?
miss.dx <- dplyr::anti_join(mri.info, dx.redux, by = c("RID", "VISCODE") )
# miss.dx
# How many have MRI but not diagnosis?
nrow(miss.dx)
```

```
## [1] 9
```

```
# variable names for cortical thickness measures
keep.thick <- c("ST13TA", "ST14TA", "ST15TA", "ST23TA", "ST24TA", "ST25TA",
                "ST26TA", "ST31TA", "ST32TA", "ST129TA", "ST34TA", "ST35TA",
                "ST36TA", "ST38TA", "ST39TA", "ST40TA", "ST43TA", "ST44TA",
                "ST45TA", "ST46TA", "ST47TA", "ST48TA", "ST49TA", "ST50TA",
                "ST51TA", "ST52TA", "ST54TA", "ST55TA", "ST56TA", "ST57TA",
                "ST58TA", "ST59TA", "ST60TA", "ST62TA", "ST72TA", "ST73TA",
                "ST74TA", "ST82TA", "ST83TA", "ST84TA", "ST85TA", "ST90TA",
                "ST91TA", "ST130TA", "ST93TA", "ST94TA", "ST95TA", "ST97TA",
                "ST98TA", "ST99TA", "ST102TA", "ST103TA", "ST104TA", "ST105TA",
                "ST106TA", "ST107TA", "ST108TA", "ST109TA","ST110TA", "ST111TA",
                "ST113TA", "ST114TA", "ST115TA", "ST116TA", "ST117TA","ST118TA",
                "ST119TA", "ST121TA"
)

# data sets for analysis
ddf <- dplyr::left_join(dx.redux, demo,
             by = c("RID", "VISCODE"))
thick.df <- merge(ddf, ucsffsx6[, c(keep.all, keep.imd, keep.thick)],
                  by = c("RID", "VISCODE"))

# only keep subjects who passed quality control
thick.dfp <- thick.df[thick.df$OVERALLQC == "Pass", ]
```

As mentioned above, some subjects have multiple entries in `dxsum` for the same visit. There are several situations, some of which have differing implications.

(1) There are multiple entries in `dxsum` and 1 entry in `ucsffsx6`, but while the `EXAMDATE` differs, the diagnosis does not. We can just delete an entry in such case. (RID's are 1052 (1st data frame), 4214 (3rd data frame), 4576 (5th data frame), 4643 (6th data frame).)

(2) There are multiple entries in `dxsum` and 1 entry in `ucsffsx6`, but while `EXAMDATE` the `DIAGNOSIS` differs between entries. It seems ever so slightly more principled to take the most recent `EXAMDATE`. (RID's are 4199 (2nd data frame))

(3) RID 6014 (7th data frame) appears to just be repeated, which arises from `dxsum`.\footnote{Type `dxsum[dxsum$RID == 6014, ]` into the console.} One of these can just be deleted.

(4) RID 4354 is quite odd; this subject has 2 `EXAMDATE` entries for month 84 and also has 2 unique MRI scans. Is this a data entry error? The times are close enough that it seems doubtful it was already time for month 96 visit. Is the data entry error actually that the ID is incorrect? For now, we are removing the subject entirely.

```
# run this code to see duplicates
for (i in 1:length(unique(thick.dfp$RID))) {
  curr <- thick.dfp[thick.dfp$RID == unique(thick.dfp$RID)[i], ]
  if (length(unique(curr$VISCODE)) != length(curr$VISCODE)) {
    print(curr[, 1:10])
  }
}
```

```
# remove various kinds of duplicate entries
rm.thick <- c("5", "142", "152", "189", "190", "191", "192", "258", "273", "438")

# all data
thick.dfp <- thick.dfp[!(rownames(thick.dfp) %in% rm.thick), ]

# datdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/data/"
# saveRDS(thick.dfp,
#         file = paste0(datdir, "thick_df.rds")
# )

# just screening visit (i.e., baseline)
thick.sc <- thick.dfp[thick.dfp$VISCODE == "sc", ]
# saveRDS(thick.sc,
#         file = paste0(datdir, "thick_sc.rds")
# )
```

2.2 Tau PET

As with cortical thickness, we only retain subjects who fully passed quality control, remove any subjects who have MRI but lack a diagnosis, removed several duplicate entries, and used screening visit (baseline) as the cross section. We can obtain variable names set the Tau PET data as follows:

```
# get variable names for tau PET
data("ucberkeleyav1451")
tau.pet.names <- names(ucberkeleyav1451[grep(pattern = 'CTX.*SUVR',
```

```
                                                x = names(ucberkeleyav1451),
                                                perl = TRUE)])

tau.df <- merge(ddf, ucberkeleyav1451[, c(keep.all, tau.pet.names)],
                by = c("RID", "VISCODE"))
```

As with cortical thickness, the tau PET has some subjects with various duplicates.

(1) RID's 4175 (1st data frame), 4576 (3rd data frame), 4643 (4th data frame) have multiple `EXAMDATE` for the same visit, but `DIAGNOSIS` is the same. We just delete one of them.

(2) RID 4199 (2nd data frame) has differing diagnoses for month 84. We chose the most recent.

(3) RID 6014, as before, appears to have simply been entered twice. We just delete one of them.

```
# run this code to see duplicates
for (i in 1:length(unique(tau.df$RID))) {
  curr <- tau.df[tau.df$RID == unique(tau.df$RID)[i], ]
  if (length(unique(curr$VISCODE)) != length(curr$VISCODE)) {
    print(curr[, 1:10])
  }
}
```

```
# Remove subjects with various kinds of duplicate entries
rm.tau <- c("130", "144", "276", "302", "482")

# all data
tau.df <- tau.df[!(rownames(tau.df) %in% rm.tau), ]

# datdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/data/"
# saveRDS(tau.df, file = paste0(datdir, "tau_df.rds"))

# just baseline
tau.bl <- tau.df[tau.df$VISCODE == "bl", ]
# saveRDS(tau.bl, file = paste0(datdir, "tau_bl.rds"))
```

3 Standardization

Penalized models are typically standardized. One reason for standardization is so that predictors will be on the same scale and thus important predictors will not be ommitted (or irrelevant ones kept) simply due to being on different scales. Typically, standardization is to mean 0 and standard deviation 1. However, the standard deviation for the predictors used here was already less than 1, and standardizing to standard deviation of 1 apparently degraded performance in some projects (perhaps to due to artificial inflation of the variance?) It thus made sense to scale to a value of similar order to the data. You can uncomment `thick.sd` or `tau.sd` to see all the standard deviations for each variable, but for both sets the mean standard deviation value was 0.20. We therefore scaled the predictors to mean 0 and standard deviation 0.20, which resulted in similar performance to the unscaled approach, but seems more in line with standard application of penalized regression models.

```
options(scipen = 999)
# b/c setwd() is weird in rmarkdown and the file location names are too long.
datdir <- "C:/Users/cotto/Documents/Publications/Rcode-ADNI/data/"

# cortical thickness data for various visits
thick.sc <- readRDS(
   file = paste0(datdir, "thick_sc.rds")
   )

# tau data for various visits

tau.bl <- readRDS(
   file = paste0(datdir, "tau_bl.rds")
   )

thick.sd <- apply(thick.sc[ , names(thick.sc[grep(pattern = 'ST.*TA',
                                                  x = names(thick.sc),
                                                  perl = TRUE)])],
                  2, sd, na.rm = TRUE)
#thick.sd

tau.sd <- apply(tau.bl[ , names(tau.bl[grep(pattern = 'CTX.*SUVR',
                                                  x = names(tau.bl),
                                                  perl = TRUE)])],
                2, sd, na.rm = TRUE)
#tau.sd

mean(thick.sd)
```

```
## [1] 0.1976702
```

```
sd(thick.sd)
```

```
## [1] 0.07403171
```

```
mean(tau.sd)
```

```
## [1] 0.1970015
```

```
sd(tau.sd)
```

```
## [1] 0.05635141
```

4 Neighborhood Matrix Generation

It may be relevant to understand how the neighborhood matrix was generated/chosen for the IAR priors. A few comments/clarifications:

(1) We assume that if 2 regions of the Desikan-Killiany Atlas "touch" (share a "border") then they are neighbors.

(2) The labels must account for the fact that we have 2 hemispheres. We assume that neighbors are only within a given hemisphere.

(3) It is possible that a better scheme for choosing neighbors could be devised, but this is beyond what we want to explore in this work. Perhaps it is a limitation of the paper.

Below are the names of each region of the Desikan-Killiany Atlas:

```
region.names <- c(
  "left.bank", "left.caudal.anterior.cingulate", "left.caudal.middle.frontal",
  "left.cuneus", "left.entorhinal", "left.frontal.pole", "left.fusiform",
  "left.inferior.parietal", "left.inferior.temporal", "left.insula",
  "left.isthmus.cingulate", "left.lateral.occipital", "left.lateral.orbitofrontal",
  "left.lingual", "left.medial.orbitofrontal", "left.middle.temporal",
  "left.paracentral", "left.parahippocampal",  "left.pars.opercularis",
  "left.pars.orbitalis", "left.pars.triangularis", "left.pericalcarine",
  "left.postcentral", "left.posterior.cingulate", "left.precentral",
  "left.precuneus", "left.rostral.anterior.cingulate", "left.rostral.middle.frontal",
  "left.superior.frontal", "left.superior.parietal", "left.superior.temporal",
  "left.supramarginal", "left.temporal.pole", "left.transverse.temporal",
  "right.bank", "right.caudal.anterior.cingulate", "right.caudal.middle.frontal",
  "right.cuneus", "right.entorhinal", "right.frontal.pole", "right.fusiform",
  "right.inferior.parietal", "right.inferior.temporal", "right.insula",
  "right.isthmus.cingulate", "right.lateral.occipital", "right.lateral.orbitofrontal",
  "right.lingual", "right.medial.orbitofrontal", "right.middle.temporal",
  "right.paracentral", "right.parahippocampal", "right.pars.opercularis",
  "right.pars.orbitalis", "right.pars.triangularis", "right.pericalcarine",
  "right.postcentral", "right.posterior.cingulate", "right.precentral",
  "right.precuneus", "right.rostral.anterior.cingulate", "right.rostral.middle.frontal",
  "right.superior.frontal", "right.superior.parietal", "right.superior.temporal",
  "right.supramarginal", "right.temporal.pole", "right.transverse.temporal"
                  )
```

Below are the variable names for mean cortical thickness in each region:

```
thick.mean <- c(
  "ST13TA", "ST14TA", "ST15TA", "ST23TA", "ST24TA", "ST25TA", "ST26TA", "ST31TA",
  "ST32TA", "ST129TA", "ST34TA", "ST35TA", "ST36TA", "ST38TA", "ST39TA", "ST40TA",
  "ST43TA", "ST44TA", "ST45TA", "ST46TA", "ST47TA",  "ST48TA", "ST49TA", "ST50TA",
  "ST51TA", "ST52TA", "ST54TA", "ST55TA", "ST56TA", "ST57TA", "ST58TA", "ST59TA",
  "ST60TA", "ST62TA", "ST72TA", "ST73TA", "ST74TA", "ST82TA", "ST83TA", "ST84TA",
  "ST85TA", "ST90TA", "ST91TA", "ST130TA", "ST93TA", "ST94TA", "ST95TA", "ST97TA",
  "ST98TA", "ST99TA", "ST102TA", "ST103TA", "ST104TA", "ST105TA", "ST106TA", "ST107TA",
  "ST108TA", "ST109TA", "ST110TA", "ST111TA", "ST113TA", "ST114TA", "ST115TA", "ST116TA",
  "ST117TA", "ST118TA", "ST119TA", "ST121TA"
                )
```

Below are the variable names for the Tau PET data:

```
tau.pet <- c(
  "CTX_LH_BANKSSTS_SUVR", "CTX_LH_CAUDALANTERIORCINGULATE_SUVR",
  "CTX_LH_CAUDALMIDDLEFRONTAL_SUVR", "CTX_LH_CUNEUS_SUVR",
  "CTX_LH_ENTORHINAL_SUVR", "CTX_LH_FRONTALPOLE_SUVR",
```

```
  "CTX_LH_FUSIFORM_SUVR", "CTX_LH_INFERIORPARIETAL_SUVR",
  "CTX_LH_INFERIORTEMPORAL_SUVR", "CTX_LH_INSULA_SUVR",
  "CTX_LH_ISTHMUSCINGULATE_SUVR", "CTX_LH_LATERALOCCIPITAL_SUVR",
  "CTX_LH_LATERALORBITOFRONTAL_SUVR", "CTX_LH_LINGUAL_SUVR",
  "CTX_LH_MEDIALORBITOFRONTAL_SUVR", "CTX_LH_MIDDLETEMPORAL_SUVR",
  "CTX_LH_PARACENTRAL_SUVR", "CTX_LH_PARAHIPPOCAMPAL_SUVR",
  "CTX_LH_PARSOPERCULARIS_SUVR", "CTX_LH_PARSORBITALIS_SUVR",
  "CTX_LH_PARSTRIANGULARIS_SUVR", "CTX_LH_PERICALCARINE_SUVR",
  "CTX_LH_POSTCENTRAL_SUVR", "CTX_LH_POSTERIORCINGULATE_SUVR",
  "CTX_LH_PRECENTRAL_SUVR", "CTX_LH_PRECUNEUS_SUVR",
  "CTX_LH_ROSTRALANTERIORCINGULATE_SUVR", "CTX_LH_ROSTRALMIDDLEFRONTAL_SUVR",
  "CTX_LH_SUPERIORFRONTAL_SUVR", "CTX_LH_SUPERIORPARIETAL_SUVR",
  "CTX_LH_SUPERIORTEMPORAL_SUVR", "CTX_LH_SUPRAMARGINAL_SUVR",
  "CTX_LH_TEMPORALPOLE_SUVR", "CTX_LH_TRANSVERSETEMPORAL_SUVR",
  "CTX_RH_BANKSSTS_SUVR",
  "CTX_RH_CAUDALANTERIORCINGULATE_SUVR", "CTX_RH_CAUDALMIDDLEFRONTAL_SUVR",
  "CTX_RH_CUNEUS_SUVR", "CTX_RH_ENTORHINAL_SUVR",
  "CTX_RH_FRONTALPOLE_SUVR", "CTX_RH_FUSIFORM_SUVR",
  "CTX_RH_INFERIORPARIETAL_SUVR", "CTX_RH_INFERIORTEMPORAL_SUVR",
  "CTX_RH_INSULA_SUVR", "CTX_RH_ISTHMUSCINGULATE_SUVR",
  "CTX_RH_LATERALOCCIPITAL_SUVR", "CTX_RH_LATERALORBITOFRONTAL_SUVR",
  "CTX_RH_LINGUAL_SUVR", "CTX_RH_MEDIALORBITOFRONTAL_SUVR",
  "CTX_RH_MIDDLETEMPORAL_SUVR", "CTX_RH_PARACENTRAL_SUVR",
  "CTX_RH_PARAHIPPOCAMPAL_SUVR", "CTX_RH_PARSOPERCULARIS_SUVR",
  "CTX_RH_PARSORBITALIS_SUVR", "CTX_RH_PARSTRIANGULARIS_SUVR",
  "CTX_RH_PERICALCARINE_SUVR", "CTX_RH_POSTCENTRAL_SUVR",
  "CTX_RH_POSTERIORCINGULATE_SUVR", "CTX_RH_PRECENTRAL_SUVR",
  "CTX_RH_PRECUNEUS_SUVR", "CTX_RH_ROSTRALANTERIORCINGULATE_SUVR",
  "CTX_RH_ROSTRALMIDDLEFRONTAL_SUVR", "CTX_RH_SUPERIORFRONTAL_SUVR",
  "CTX_RH_SUPERIORPARIETAL_SUVR", "CTX_RH_SUPERIORTEMPORAL_SUVR",
  "CTX_RH_SUPRAMARGINAL_SUVR", "CTX_RH_TEMPORALPOLE_SUVR",
  "CTX_RH_TRANSVERSETEMPORAL_SUVR"
)
```

Now we specify the neighbors for each region:

```
variable.labels <- data.frame(region.id = 1:68, region.names, thick.mean, tau.pet)

dsa.list <- list(
 left.bank = c(
   "left.inferior.parietal", "left.superior.temporal", "left.middle.temporal"),
 left.caudal.anterior.cingulate = c(
   "left.rostral.anterior.cingulate", "left.superior.frontal",
   "left.posterior.cingulate"),
 left.caudal.middle.frontal = c(
   "left.superior.frontal", "left.rostral.middle.frontal",
    "left.precentral", "left.pars.opercularis"),
 left.cuneus = c(
   "left.precuneus", "left.pericalcarine",
   "left.superior.parietal", "left.lateral.occipital"),
 left.entorhinal = c(
   "left.temporal.pole", "left.parahippocampal",
   "left.fusiform"),
```

```
left.frontal.pole = c(
  "left.rostral.middle.frontal", "left.pars.orbitalis",
  "left.medial.orbitofrontal"),
left.fusiform = c(
  "left.temporal.pole", "left.entorhinal", "left.parahippocampal",
  "left.lingual", "left.lateral.occipital", "left.inferior.temporal"),
left.inferior.parietal = c(
  "left.superior.parietal", "left.supramarginal", "left.superior.temporal",
  "left.bank", "left.middle.temporal", "left.lateral.occipital"),
left.inferior.temporal = c(
  "left.middle.temporal", "left.lateral.occipital", "left.fusiform"),
left.insula <- c(
  "left.lateral.orbitofrontal", "left.pars.triangularis",
  "left.pars.opercularis", "left.precentral", "left.postcentral",
  "left.supramarginal", "left.transverse.temporal", "left.superior.temporal"),
left.isthmus.cingulate = c(
  "left.precuneus", "left.lingual", "left.parahippocampal",
  "left.posterior.cingulate"),
left.lateral.occipital = c(
  "left.inferior.parietal", "left.superior.parietal", "left.fusiform",
  "left.inferior.temporal",  "left.cuneus", "left.pericalcarine",
  "left.lingual"),
left.lateral.orbitofrontal = c(
  "left.insula", "left.pars.orbitalis", "left.medial.orbitofrontal",
  "left.pars.triangularis"),
left.lingual = c(
  "left.precuneus", "left.pericalcarine", "left.lateral.occipital",
  "left.fusiform", "left.parahippocampal", "left.isthmus.cingulate"),
left.medial.orbitofrontal = c(
  "left.lateral.orbitofrontal", "left.rostral.anterior.cingulate",
  "left.superior.frontal", "left.rostral.middle.frontal"),
left.middle.temporal = c(
  "left.bank", "left.inferior.temporal", "left.superior.temporal",
  "left.inferior.parietal", "left.lateral.occipital"),
left.paracentral = c(
  "left.superior.frontal", "left.posterior.cingulate",
  "left.precuneus","left.precentral", "left.postcentral"),
left.parahippocampal = c(
  "left.entorhinal", "left.fusiform",  "left.lingual",
  "left.isthmus.cingulate"),
left.pars.opercularis = c(
  "left.insula", "left.pars.triangularis", "left.rostral.middle.frontal",
  "left.caudal.middle.frontal", "left.precentral"),
left.pars.orbitalis = c(
  "left.lateral.orbitofrontal", "left.rostral.middle.frontal",
  "left.pars.triangularis", "left.frontal.pole"),
left.pars.triangularis = c(
  "left.insula", "left.pars.opercularis", "left.rostral.middle.frontal",
  "left.pars.orbitalis"),
left.pericalcarine = c(
  "left.cuneus", "left.lingual", "left.lateral.occipital", "left.precuneus"),
left.postcentral = c(
  "left.insula", "left.superior.parietal", "left.supramarginal",
```

```
  "left.precentral", "left.paracentral"),
left.posterior.cingulate = c(
  "left.paracentral", "left.caudal.anterior.cingulate", "left.isthmus.cingulate",
  "left.precuneus", "left.superior.frontal"),
left.precentral = c(
  "left.insula", "left.postcentral", "left.caudal.middle.frontal",
  "left.pars.opercularis", "left.superior.frontal","left.paracentral"),
left.precuneus = c(
  "left.cuneus", "left.paracentral", "left.posterior.cingulate",
  "left.isthmus.cingulate", "left.lingual", "left.pericalcarine"),
left.rostral.anterior.cingulate = c(
  "left.caudal.anterior.cingulate", "left.superior.frontal",
  "left.medial.orbitofrontal"),
left.rostral.middle.frontal = c(
  "left.pars.triangularis", "left.pars.orbitalis", "left.caudal.middle.frontal",
  "left.pars.opercularis", "left.superior.frontal", "left.frontal.pole",
  "left.medial.orbitofrontal"),
left.superior.frontal = c(
  "left.paracentral", "left.posterior.cingulate", "left.caudal.anterior.cingulate",
  "left.rostral.anterior.cingulate", "left.medial.orbitofrontal",
  "left.rostral.middle.frontal", "left.caudal.middle.frontal", "left.precentral"),
left.superior.parietal = c(
  "left.postcentral", "left.supramarginal", "left.inferior.parietal",
  "left.lateral.occipital", "left.precuneus", "left.cuneus"),
left.superior.temporal = c(
  "left.insula", "left.supramarginal",  "left.inferior.parietal",
  "left.middle.temporal", "left.bank", "left.transverse.temporal"),
left.supramarginal = c(
  "left.insula", "left.superior.parietal", "left.postcentral",
  "left.superior.temporal", "left.inferior.parietal", "left.transverse.temporal"),
left.temporal.pole = c(
  "left.entorhinal", "left.fusiform", "left.superior.temporal"),
left.transverse.temporal = c(
  "left.insula", "left.superior.temporal", "left.supramarginal"),
right.bank = c(
  "right.inferior.parietal",  "right.superior.temporal", "right.middle.temporal"),
right.caudal.anterior.cingulate = c(
  "right.rostral.anterior.cingulate", "right.superior.frontal",
  "right.posterior.cingulate"),
right.caudal.middle.frontal = c(
  "right.superior.frontal", "right.rostral.middle.frontal", "right.precentral",
  "right.pars.opercularis"),
right.cuneus = c(
  "right.precuneus", "right.pericalcarine", "right.superior.parietal",
  "right.lateral.occipital"),
right.entorhinal = c(
  "right.temporal.pole", "right.parahippocampal", "right.fusiform"),
right.frontal.pole = c(
  "right.rostral.middle.frontal", "right.pars.orbitalis",
  "right.medial.orbitofrontal"),
right.fusiform = c(
  "right.temporal.pole", "right.entorhinal", "right.parahippocampal",
  "right.lingual", "right.lateral.occipital", "right.inferior.temporal"),
```

```
right.inferior.parietal = c(
  "right.superior.parietal", "right.supramarginal", "right.superior.temporal",
  "right.bank", "right.middle.temporal", "right.lateral.occipital"),
right.inferior.temporal = c(
  "right.middle.temporal", "right.lateral.occipital", "right.fusiform"),
right.insula <- c(
  "right.lateral.orbitofrontal", "right.pars.triangularis", "right.pars.opercularis",
  "right.precentral", "right.postcentral", "right.supramarginal",
  "right.transverse.temporal", "right.superior.temporal"),
right.isthmus.cingulate = c(
  "right.precuneus", "right.lingual", "right.parahippocampal",
  "right.posterior.cingulate"),
right.lateral.occipital = c(
  "right.inferior.parietal", "right.superior.parietal", "right.fusiform",
  "right.inferior.temporal", "right.cuneus", "right.pericalcarine",
  "right.lingual"),
right.lateral.orbitofrontal = c(
  "right.insula", "right.pars.orbitalis",
  "right.medial.orbitofrontal", "right.pars.triangularis"),
right.lingual = c(
  "right.precuneus", "right.pericalcarine", "right.lateral.occipital",
  "right.fusiform", "right.parahippocampal", "right.isthmus.cingulate"),
right.medial.orbitofrontal = c(
  "right.lateral.orbitofrontal", "right.rostral.anterior.cingulate",
  "right.superior.frontal", "right.rostral.middle.frontal"),
right.middle.temporal = c(
  "right.bank", "right.inferior.temporal", "right.superior.temporal",
  "right.inferior.parietal", "right.lateral.occipital"),
right.paracentral = c(
  "right.superior.frontal", "right.posterior.cingulate", "right.precuneus",
  "right.precentral","right.postcentral"),
right.parahippocampal = c(
  "right.entorhinal", "right.fusiform", "right.lingual",
  "right.isthmus.cingulate"),
right.pars.opercularis = c(
  "right.insula", "right.pars.triangularis", "right.rostral.middle.frontal",
  "right.caudal.middle.frontal", "right.precentral"),
right.pars.orbitalis = c(
  "right.lateral.orbitofrontal", "right.rostral.middle.frontal",
  "right.pars.triangularis", "right.frontal.pole"),
right.pars.triangularis = c(
  "right.insula", "right.pars.opercularis", "right.rostral.middle.frontal",
  "right.pars.orbitalis"),
right.pericalcarine = c(
  "right.cuneus", "right.lingual", "right.lateral.occipital",
   "right.precuneus"),
right.postcentral = c(
  "right.insula", "right.superior.parietal", "right.supramarginal",
  "right.precentral", "right.paracentral"),
right.posterior.cingulate = c(
  "right.paracentral", "right.caudal.anterior.cingulate", "right.isthmus.cingulate",
  "right.precuneus", "right.superior.frontal"),
right.precentral = c(
```

```
   "right.insula", "right.postcentral", "right.caudal.middle.frontal",
   "right.pars.opercularis", "right.superior.frontal", "right.paracentral"),
 right.precuneus = c(
   "right.cuneus", "right.paracentral", "right.posterior.cingulate",
   "right.isthmus.cingulate", "right.lingual", "right.pericalcarine"),
 right.rostral.anterior.cingulate = c(
   "right.caudal.anterior.cingulate", "right.superior.frontal",
   "right.medial.orbitofrontal"),
 right.rostral.middle.frontal = c(
   "right.pars.triangularis", "right.pars.orbitalis", "right.caudal.middle.frontal",
    "right.pars.opercularis", "right.superior.frontal",  "right.frontal.pole",
    "right.medial.orbitofrontal"),
 right.superior.frontal = c(
   "right.paracentral", "right.posterior.cingulate", "right.caudal.anterior.cingulate",
   "right.rostral.anterior.cingulate",  "right.medial.orbitofrontal",
   "right.rostral.middle.frontal", "right.caudal.middle.frontal", "right.precentral"),
 right.superior.parietal = c(
   "right.postcentral", "right.supramarginal", "right.inferior.parietal",
   "right.lateral.occipital", "right.precuneus", "right.cuneus"),
 right.superior.temporal = c(
   "right.insula", "right.supramarginal", "right.inferior.parietal",
   "right.middle.temporal", "right.bank", "right.transverse.temporal"),
 right.supramarginal = c(
   "right.insula", "right.superior.parietal", "right.postcentral", "right.superior.temporal",
   "right.inferior.parietal", "right.transverse.temporal"),
 right.temporal.pole = c(
   "right.entorhinal", "right.fusiform", "right.superior.temporal"),
 right.transverse.temporal = c(
   "right.insula", "right.superior.temporal", "right.supramarginal")
 )

dsa.list.c <- c()
for (i in 1:length(dsa.list)) {
  dsa.list.c <- c(dsa.list.c, dsa.list[[i]])
}

# Verify we have every region accounted for
length(unique(dsa.list.c))
```

```
## [1] 68
```

With the neighbors specified we can create a neighborhood matrix:

```
sparse.nb <- data.frame(location.index = NULL, nb.index = NULL)

for (i in 1:length(region.names)) {
  dsi <- dsa.list[[i]]
  for (j in 1:length(dsi)) {
    nb.ij <- data.frame(location.index = i, nb.index = which(region.names == dsi[j]))
    sparse.nb <- rbind(sparse.nb, nb.ij)
  }
}
```

```
sparse.nb$weights <- 1

knitr::kable(
  head(sparse.nb, 25)
)
```

location.index	nb.index	weights
1	8	1
1	31	1
1	16	1
2	27	1
2	29	1
2	24	1
3	29	1
3	28	1
3	25	1
3	19	1
4	26	1
4	22	1
4	30	1
4	12	1
5	33	1
5	18	1
5	7	1
6	28	1
6	20	1
6	15	1
7	33	1
7	5	1
7	18	1
7	14	1
7	12	1

We now prepare this information for modeling with IAR priors in the `ssnet` package. It is recommended to save the file and not have to mess with it any further.

```
library(ssnet)

dk.nb <- mungeCARdata4stan_irregular(sparse.nb$nb.index, table(sparse.nb$location.index))

dk.nb
```

```
## $J
## [1] 68
##
## $J_edges
## [1] 162
##
## $node1
##   [1]  1  1  1  2  2  2  3  3  3  3  4  4  4  4  5  5  5  6  6  6  7  7  7  7  7
##  [26]  8  8  8  8  8  9  9 10 10 10 10 10 10 10 10 11 11 11 11 12 12 12 13 13 13
```

```
##  [51] 14 14 14 15 15 15 16 17 17 17 17 17 19 19 19 20 20 21 22 23 23 23 24 24 25
##  [76] 27 28 30 31 31 32 35 35 35 36 36 36 37 37 37 37 38 38 38 38 39 39 39 40 40
## [101] 40 41 41 41 41 41 42 42 42 42 42 43 43 44 44 44 44 44 44 44 44 45 45 45 45
## [126] 46 46 46 47 47 47 48 48 48 49 49 49 50 51 51 51 51 51 53 53 53 54 54 55 56
## [151] 57 57 57 58 58 59 61 62 64 65 65 66
##
## $node2
##   [1]  8 31 16 27 29 24 29 28 25 19 26 22 30 12 33 18  7 28 20 15 33 18 14 12  9
##  [26] 30 32 31 16 12 16 12 13 21 19 25 23 32 34 31 26 14 18 24 30 22 14 20 15 21
##  [51] 26 22 18 27 29 28 31 29 24 26 25 23 21 28 25 28 21 28 26 30 32 25 26 29 29
##  [76] 29 29 32 32 34 34 42 65 50 61 63 58 63 62 59 53 60 56 64 46 67 52 41 62 54
## [101] 49 67 52 48 46 43 64 66 65 50 46 50 46 47 55 53 59 57 66 68 65 60 48 52 58
## [126] 64 56 48 54 49 55 60 56 52 61 63 62 65 63 58 60 59 57 55 62 59 62 55 62 60
## [151] 64 66 59 60 63 63 63 63 66 66 68 68
```

```
saveRDS(dk.nb, "data/dk_nb.rds")

# datdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/data/"
# saveRDS(dk.nb, file = paste0(datdir, "dk_nb.rds"))
```