Supplementary Materials: Details on Data, Analyses, and Reproduction of Tables and Figures

Justin M. Leach

1 Introduction

This rmarkdown produces relevant `R` code to re-produce analyses in the paper "The Spike-and-Slab Elastic Net as a Classification Tool in Alzheimer's Disease." Topics include:

(1) How the data were cleaned/processed after obtaining from ADNI.
(2) How the data were analyzed.
(3) Reproducing plots and tables from the paper.

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. 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 2020-04-12.

```
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")
```

```
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] 16
```

```
# 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 Data Analysis

3.1 Relevant Information

The data analyses utilizes the `R` package `ssnet`, which is available at https://github.com/jmleach-bst/ssnet. The version used is 0.0.0.9000; the primary dependencies are `BhGLM` (version 1.1.0) and `rstan` (version 2.19.2). You may also need `sim2Dpredictr` (version 0.1.0), which is available at https://github.com/jmleach-bst/sim2Dpredictr. The analyses were performed using research computing at the University of Alabama at Birmingham (Cheaha). Within the same folder you used to open this current Rmd file you can find `R` scripts contained in the folder "Rcode_analyses_cheaha" and shell scripts needed to run the jobs on slurm in the folder "scripts_analyses_cheaha". Note that these files cannot be used without edit to reproduce the analyses; you'll need to ensure the code in each file specifyies the correct directories (you have to choose these!) for accessing data (wherever you've saved it!) and saving results (wherever you want it!). Windows can be somewhat difficult about opening the shell scripts, so you may want to use the "Open With" option and choose your editor of choice (I like Notepad++, but use whatever works for you). If you would rather not re-run all the files, the results are found in the "results_analyses_cheaha_scaled_2" folder.

Note: We lost some files through accidental deletion. Fortunately, the original version of this file was backed up. The files we lost explored how results changed based on scaling the predictors. The basic point is that when using shrinkage methods it is preferable to center and scale (standardize) all the predictors to have the same mean and variance. 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 (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/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/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
```

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

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

3.2 Neighborhood Matrix Generation

It is 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:

```r
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:

```r
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

head(sparse.nb)
```

```
##   location.index nb.index weights
## 1              1        8       1
## 2              1       31       1
## 3              1       16       1
## 4              2       27       1
## 5              2       29       1
## 6              2       24       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
```

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

4 Reproducing Tables

Obviously you'll need to alter to file names to match locations on your machine. You'll also need the "helper functions" `pred.smry`.[1]

4.1 Cortical Thickness

4.1.1 CN vs. Dementia

Top sections of Tables 1 and 2 (CN vs. Dementia).

```
resdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/results_analys
# load results under EN parameter alpha = 1 (Lasso)
sc.cn.d.l <- readRDS(
  file = paste0(resdir, "thickness_sc_cn_d_lasso_grid.rds")
)

# optimal choices of s0 & s1 under EN parameter alpha = 1 (Lasso)
sc.cn.d.l.opt <- pred.smry(model.fit = sc.cn.d.l,
                           reduced.data = sc.cn.d,
                           x.prefix = "ST.*",
                           classify = FALSE)

# load results under EN parameter alpha = 0.5
sc.cn.d.en <- readRDS(
  file = paste0(resdir, "thickness_sc_cn_d_en_grid.rds")
)

# optimal choices of s0 & s1 under EN parameter alpha = 0.5
sc.cn.d.en.opt <- pred.smry(model.fit = sc.cn.d.en,
                           reduced.data = sc.cn.d,
                           x.prefix = "ST.*",
                           classify = FALSE)

cn.d.smry <- rbind(sc.cn.d.l.opt,
                   sc.cn.d.en.opt)
cn.d.smry[, -1] <- round(cn.d.smry[, -1], digits = 3)
```

4.1.2 CN vs. MCI

Middle sections of Tables 1 and 2 (CN vs. MCI).

[1] The code for this function is in Rmd file, but the output is omitted in PDF to reduce clutter.

```
resdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/results_analys
# load results under EN parameter alpha = 1 (Lasso)
sc.cn.mci.l <- readRDS(
  file = paste0(resdir, "thickness_sc_cn_mci_lasso_grid.rds")
  )

# optimal choices of s0 & s1 under EN parameter alpha = 1 (Lasso)
sc.cn.mci.l.opt <- pred.smry(model.fit = sc.cn.mci.l,
                             reduced.data = sc.cn.mci,
                             x.prefix = "ST.*",
                             classify = FALSE)

# load results under EN parameter alpha = 0.5
sc.cn.mci.en <- readRDS(
  file = paste0(resdir, "thickness_sc_cn_mci_en_grid.rds")
  )

# optimal choices of s0 & s1 under EN parameter alpha = 0.5
sc.cn.mci.en.opt <- pred.smry(model.fit = sc.cn.mci.en,
                              reduced.data = sc.cn.mci,
                              x.prefix = "ST.*",
                              classify = FALSE)

cn.mci.smry <- rbind(sc.cn.mci.l.opt,
                     sc.cn.mci.en.opt)
cn.mci.smry[, -1] <- round(cn.mci.smry[, -1], digits = 3)
```

4.1.3 MCI vs. Dementia

Bottom sections of Tables 1 and 2 (MCI vs. Dementia).

```
resdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing//Rcode-ADNI//results_analy
# load results under EN parameter alpha = 1 (Lasso)
sc.mci.d.l <- readRDS(
  file = paste0(resdir, "thickness_sc_mci_d_lasso_grid.rds")
  )

# optimal choices of s0 & s1 under EN parameter alpha = 1 (Lasso)
sc.mci.d.l.opt <- pred.smry(model.fit = sc.mci.d.l,
                            reduced.data = sc.mci.d,
                            x.prefix = "ST.*",
                           classify = FALSE)

# load results under EN parameter alpha = 0.5
sc.mci.d.en <- readRDS(
  file = paste0(resdir, "thickness_sc_mci_d_en_grid.rds")
  )

# optimal choices of s0 & s1 under EN parameter alpha = 0.5
sc.mci.d.en.opt <- pred.smry(model.fit = sc.mci.d.en,
                             reduced.data = sc.mci.d,
                             x.prefix = "ST.*",
```

```
                              classify = FALSE)

mci.d.smry <- rbind(sc.mci.d.l.opt,
                    sc.mci.d.en.opt)
mci.d.smry[, -1] <- round(mci.d.smry[, -1], digits = 3)
```

4.1.4 Table Generation

```
ct.smry <- cbind(
  outcome = c(rep("CN vs. Dem.", 6),
              rep("CN vs. MCI", 6),
              rep("MCI vs. Dem.", 6)),
  rbind(cn.d.smry,
        cn.mci.smry,
        mci.d.smry)
)

# saveRDS(ct.smry,
#         file = paste0(resdir, "ct_smry.rds"))

# Table 1 in paper
knitr::kable(ct.smry %>% select("outcome", "model", "alpha", "s0", "s1",
                                "deviance", "auc", "mse", "mae",
                                "misclassification"),
             col.names = c("Outcome", "Model", "alpha", "s0", "s1",
                           "Dev.", "AUC", "MSE", "MAE", "Misclass."),
             caption = "Cortical Thickness: Prediction Error Estimates")
```

Table 1: Cortical Thickness: Prediction Error Estimates

Outcome	Model	alpha	s0	s1	Dev.	AUC	MSE	MAE	Misclass.
CN vs. Dem.	glmnet	1.0	0.002	0.002	90.321	0.952	0.046	0.094	0.063
CN vs. Dem.	ss	1.0	0.270	7.500	73.591	0.969	0.035	0.067	0.050
CN vs. Dem.	ss_iar	1.0	0.260	6.000	70.865	0.972	0.035	0.069	0.049
CN vs. Dem.	glmnet	0.5	0.001	0.001	84.257	0.958	0.043	0.088	0.057
CN vs. Dem.	ss	0.5	0.260	10.000	71.964	0.970	0.036	0.077	0.051
CN vs. Dem.	ss_iar	0.5	0.280	10.000	67.023	0.975	0.034	0.073	0.049
CN vs. MCI	glmnet	1.0	0.007	0.007	425.341	0.622	0.208	0.414	0.289
CN vs. MCI	ss	1.0	0.150	7.000	412.024	0.665	0.198	0.389	0.279
CN vs. MCI	ss_iar	1.0	0.140	4.000	402.915	0.684	0.194	0.381	0.272
CN vs. MCI	glmnet	0.5	0.006	0.006	423.658	0.629	0.207	0.410	0.290
CN vs. MCI	ss	0.5	0.140	4.500	410.785	0.665	0.198	0.392	0.278
CN vs. MCI	ss_iar	0.5	0.150	7.500	399.505	0.694	0.192	0.377	0.276
MCI vs. Dem.	glmnet	1.0	0.009	0.009	140.894	0.790	0.148	0.293	0.210
MCI vs. Dem.	ss	1.0	0.180	7.500	123.997	0.847	0.129	0.243	0.183
MCI vs. Dem.	ss_iar	1.0	0.140	4.000	122.383	0.849	0.126	0.244	0.172
MCI vs. Dem.	glmnet	0.5	0.006	0.006	135.305	0.813	0.142	0.278	0.205
MCI vs. Dem.	ss	0.5	0.140	7.000	120.445	0.853	0.124	0.244	0.171
MCI vs. Dem.	ss_iar	0.5	0.140	5.500	119.196	0.856	0.123	0.246	0.165

```
# Table 2 in paper
knitr::kable(ct.smry %>% select("outcome", "model", "alpha", "s0", "s1",
                                "accuracy", "sensitivity",
                                "specificity", "ppv", "npv"),
             col.names = c("Outcome", "Model", "alpha", "s0", "s1",
                           "Accu.", "Sens.", "Spec.", "PPV", "NPV"),
             caption = "Cortical Thickness: Classification Performance")
```

Table 2: Cortical Thickness: Classification Performance

Outcome	Model	alpha	s0	s1	Accu.	Sens.	Spec.	PPV	NPV
CN vs. Dem.	glmnet	1.0	0.002	0.002	0.937	0.669	0.982	0.864	0.947
CN vs. Dem.	ss	1.0	0.270	7.500	0.950	0.751	0.983	0.883	0.960
CN vs. Dem.	ss_iar	1.0	0.260	6.000	0.951	0.756	0.984	0.886	0.960
CN vs. Dem.	glmnet	0.5	0.001	0.001	0.943	0.715	0.981	0.865	0.954
CN vs. Dem.	ss	0.5	0.260	10.000	0.949	0.731	0.985	0.891	0.956
CN vs. Dem.	ss_iar	0.5	0.280	10.000	0.951	0.736	0.987	0.903	0.957
CN vs. MCI	glmnet	1.0	0.007	0.007	0.711	0.197	0.966	0.744	0.708
CN vs. MCI	ss	1.0	0.150	7.000	0.721	0.329	0.915	0.656	0.733
CN vs. MCI	ss_iar	1.0	0.140	4.000	0.728	0.351	0.915	0.672	0.740
CN vs. MCI	glmnet	0.5	0.006	0.006	0.710	0.206	0.959	0.716	0.709
CN vs. MCI	ss	0.5	0.140	4.500	0.722	0.311	0.926	0.674	0.731
CN vs. MCI	ss_iar	0.5	0.150	7.500	0.724	0.358	0.905	0.652	0.740
MCI vs. Dem.	glmnet	1.0	0.009	0.009	0.790	0.344	0.941	0.659	0.810
MCI vs. Dem.	ss	1.0	0.180	7.500	0.817	0.531	0.914	0.675	0.853
MCI vs. Dem.	ss_iar	1.0	0.140	4.000	0.828	0.554	0.920	0.701	0.860
MCI vs. Dem.	glmnet	0.5	0.006	0.006	0.795	0.377	0.936	0.663	0.817
MCI vs. Dem.	ss	0.5	0.140	7.000	0.829	0.551	0.922	0.707	0.859
MCI vs. Dem.	ss_iar	0.5	0.140	5.500	0.835	0.551	0.931	0.730	0.861

4.2 Tau PET

4.2.1 CN vs. Dementia

```
resdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/results_analys
tau.bl.cn.d.l <- readRDS(
  file = paste0(resdir, "tau_bl_cn_d_lasso_grid.rds")
)

tau.bl.cn.d.l.opt <- pred.smry(model.fit = tau.bl.cn.d.l,
                        reduced.data = tau.bl.cn.d,
                        x.prefix = "CTX.*",
                        classify = FALSE)

tau.bl.cn.d.en <- readRDS(
  file = paste0(resdir, "tau_bl_cn_d_en_grid.rds")
)

tau.bl.cn.d.en.opt <- pred.smry(model.fit = tau.bl.cn.d.en,
                        reduced.data = tau.bl.cn.d,
```

```
                        x.prefix = "CTX.*",
                        classify = FALSE)

tau.bl.cn.d.smry <- rbind(tau.bl.cn.d.l.opt,
                          tau.bl.cn.d.en.opt)
tau.bl.cn.d.smry[, -1] <- round(tau.bl.cn.d.smry[, -1], digits = 3)
```

4.2.2 CN vs. MCI

```
resdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/results_analys
tau.bl.cn.mci.l <- readRDS(
  file = paste0(resdir, "tau_bl_cn_mci_lasso_grid.rds")
)

tau.bl.cn.mci.l.opt <- pred.smry(model.fit = tau.bl.cn.mci.l,
                        reduced.data = tau.bl.cn.mci,
                        x.prefix = "CTX.*",
                        classify = FALSE)

tau.bl.cn.mci.en <- readRDS(
  file = paste0(resdir, "tau_bl_cn_mci_en_grid.rds")
)

tau.bl.cn.mci.en.opt <- pred.smry(model.fit = tau.bl.cn.mci.en,
                        reduced.data = tau.bl.cn.mci,
                        x.prefix = "CTX.*",
                        classify = FALSE)

tau.bl.cn.mci.smry <- rbind(tau.bl.cn.mci.l.opt,
                          tau.bl.cn.mci.en.opt)
tau.bl.cn.mci.smry[, -1] <- round(tau.bl.cn.mci.smry[, -1], digits = 3)
```

4.2.3 MCI vs. Dementia

```
resdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/results_analys
tau.bl.mci.d.l <- readRDS(
  file = paste0(resdir, "tau_bl_mci_d_lasso_grid.rds")
)

tau.bl.mci.d.l.opt <- pred.smry(model.fit = tau.bl.mci.d.l,
                        reduced.data = tau.bl.mci.d,
                        x.prefix = "CTX.*",
                        classify = FALSE)

tau.bl.mci.d.en <- readRDS(
  file = paste0(resdir, "tau_bl_mci_d_en_grid.rds")
)

tau.bl.mci.d.en.opt <- pred.smry(model.fit = tau.bl.mci.d.en,
```

```
                        reduced.data = tau.bl.mci.d,
                        x.prefix = "CTX.*",
                        classify = FALSE)

tau.bl.mci.d.smry <- rbind(tau.bl.mci.d.l.opt,
                           tau.bl.mci.d.en.opt)
tau.bl.mci.d.smry[, -1] <- round(tau.bl.mci.d.smry[, -1], digits = 4)
```

4.2.4 Table Generation

```
tau.smry <- cbind(
  outcome = c(rep("CN vs. Dem.", 6),
              rep("CN vs. MCI", 6),
              rep("MCI vs. Dem.", 6)),
  rbind(tau.bl.cn.d.smry,
        tau.bl.cn.mci.smry,
        tau.bl.mci.d.smry)
)

# saveRDS(tau.smry,
#         file = paste0(resdir, "tau_smry.rds"))

# Table 3 in paper
knitr::kable(tau.smry %>% select("outcome", "model", "alpha", "s0", "s1",
                                 "deviance", "auc", "mse", "mae",
                                "misclassification"),
             col.names = c("Outcome", "Model", "alpha", "s0", "s1",
                           "Dev.", "AUC", "MSE", "MAE", "Misclass."),
             caption = "Tau PET: Prediction Error Estimates")
```

Table 3: Tau PET: Prediction Error Estimates

Outcome	Model	alpha	s0	s1	Dev.	AUC	MSE	MAE	Misclass.
CN vs. Dem.	glmnet	1.0	0.0020	0.0020	138.9510	0.8900	0.0630	0.1200	0.0800
CN vs. Dem.	ss	1.0	0.5000	10.0000	106.8420	0.9420	0.0510	0.1000	0.0670
CN vs. Dem.	ss_iar	1.0	0.3700	8.0000	119.7250	0.9190	0.0550	0.1040	0.0730
CN vs. Dem.	glmnet	0.5	0.0020	0.0020	134.8270	0.9030	0.0610	0.1210	0.0770
CN vs. Dem.	ss	0.5	0.2700	10.0000	118.3630	0.9190	0.0540	0.1140	0.0700
CN vs. Dem.	ss_iar	0.5	0.2700	9.5000	118.1540	0.9240	0.0540	0.1080	0.0700
CN vs. MCI	glmnet	1.0	0.0040	0.0040	443.5270	0.6830	0.1920	0.3860	0.2660
CN vs. MCI	ss	1.0	0.0600	7.5000	427.4640	0.7190	0.1820	0.3640	0.2550
CN vs. MCI	ss_iar	1.0	0.2800	5.5000	427.1450	0.7230	0.1820	0.3620	0.2560
CN vs. MCI	glmnet	0.5	0.0030	0.0030	443.6560	0.6840	0.1910	0.3820	0.2700
CN vs. MCI	ss	0.5	0.1100	10.0000	427.3200	0.7190	0.1820	0.3660	0.2570
CN vs. MCI	ss_iar	0.5	0.2500	9.0000	425.7870	0.7260	0.1810	0.3630	0.2530
MCI vs. Dem.	glmnet	1.0	0.0044	0.0044	167.9959	0.7316	0.1618	0.3112	0.2262
MCI vs. Dem.	ss	1.0	0.1700	10.0000	153.8392	0.7816	0.1438	0.2779	0.1917
MCI vs. Dem.	ss_iar	1.0	0.1700	8.5000	153.0556	0.7842	0.1430	0.2779	0.1881
MCI vs. Dem.	glmnet	0.5	0.0085	0.0085	169.3531	0.7157	0.1641	0.3218	0.2351
MCI vs. Dem.	ss	0.5	0.1700	10.0000	151.5709	0.7891	0.1425	0.2834	0.1875
MCI vs. Dem.	ss_iar	0.5	0.1700	10.0000	151.1822	0.7904	0.1421	0.2829	0.1863

```
# Table 4 in paper
knitr::kable(tau.smry %>% select("outcome", "model", "alpha", "s0", "s1",
                                 "accuracy", "sensitivity",
                                 "specificity", "ppv", "npv"),
             col.names = c("Outcome", "Model", "alpha", "s0", "s1",
                           "Accu.", "Sens.", "Spec.", "PPV", "NPV"),
             caption = "Tau PET: Classification Performance")
```

Table 4: Tau PET: Classification Performance

Outcome	Model	alpha	s0	s1	Accu.	Sens.	Spec.	PPV	NPV
CN vs. Dem.	glmnet	1.0	0.0020	0.0020	0.9200	0.5490	0.9780	0.7960	0.9320
CN vs. Dem.	ss	1.0	0.5000	10.0000	0.9330	0.6170	0.9830	0.8500	0.9420
CN vs. Dem.	ss_iar	1.0	0.3700	8.0000	0.9270	0.5900	0.9800	0.8270	0.9380
CN vs. Dem.	glmnet	0.5	0.0020	0.0020	0.9230	0.5410	0.9830	0.8310	0.9320
CN vs. Dem.	ss	0.5	0.2700	10.0000	0.9300	0.5610	0.9880	0.8790	0.9350
CN vs. Dem.	ss_iar	0.5	0.2700	9.5000	0.9300	0.5980	0.9830	0.8450	0.9400
CN vs. MCI	glmnet	1.0	0.0040	0.0040	0.7340	0.2830	0.9540	0.7470	0.7320
CN vs. MCI	ss	1.0	0.0600	7.5000	0.7450	0.3350	0.9450	0.7480	0.7450
CN vs. MCI	ss_iar	1.0	0.2800	5.5000	0.7440	0.3540	0.9340	0.7240	0.7480
CN vs. MCI	glmnet	0.5	0.0030	0.0030	0.7300	0.2760	0.9500	0.7290	0.7300
CN vs. MCI	ss	0.5	0.1100	10.0000	0.7430	0.3260	0.9460	0.7460	0.7430
CN vs. MCI	ss_iar	0.5	0.2500	9.0000	0.7470	0.3490	0.9410	0.7430	0.7480
MCI vs. Dem.	glmnet	1.0	0.0044	0.0044	0.7738	0.2195	0.9528	0.5996	0.7909
MCI vs. Dem.	ss	1.0	0.1700	10.0000	0.8083	0.3756	0.9480	0.7027	0.8247
MCI vs. Dem.	ss_iar	1.0	0.1700	8.5000	0.8119	0.3756	0.9528	0.7218	0.8254
MCI vs. Dem.	glmnet	0.5	0.0085	0.0085	0.7649	0.1780	0.9543	0.5558	0.7825
MCI vs. Dem.	ss	0.5	0.1700	10.0000	0.8125	0.3366	0.9661	0.7623	0.8186
MCI vs. Dem.	ss_iar	0.5	0.1700	10.0000	0.8137	0.3415	0.9661	0.7645	0.8197

5 Reproducing Figures

I've again commented out the `saveRDS()`, which you'll need to un-comment if reproducing analyses from scratch. Additional cleaning for figures that results in saved data are saved in the "figures_rdata" folder.

5.1 ROC Curves

5.1.1 Wrangling data

We need to wrangle the data to prepare it for ROC curves. This block does so for cortical thickness data. In retrospect, perhaps I should have written a separate function for this task.

```
# figdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/figures_rdat

sc.cn.d.l.glmnet.yhat <- apply(
  sc.cn.d.l$cv$glmnet_1$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.d.l$cv$glmnet_1$cv.estimates))),
  1, mean
```

```
)
sc.cn.d.l.ss.yhat <- apply(
  sc.cn.d.l$cv$`ss_s0=0.27_s1=7.5_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.d.l$cv$`ss_s0=0.27_s1=7.5_alpha=1`$cv.estimates))),
  1, mean
)
sc.cn.d.l.ssiar.yhat <- apply(
  sc.cn.d.l$cv$`ssiar_s0=0.26_s1=6_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.d.l$cv$`ssiar_s0=0.26_s1=6_alpha=1`$cv.estimates))),
  1, mean
)

sc.cn.d.l.yhat <- data.frame(
  model = c(rep("glmnet", length(sc.cn.d.l.glmnet.yhat)),
            rep("ss", length(sc.cn.d.l.ss.yhat)),
            rep("ss_iar", length(sc.cn.d.l.ssiar.yhat))),
  alpha = rep(1, 3 * length(sc.cn.d.l.glmnet.yhat)),
  yhat.avg = c(sc.cn.d.l.glmnet.yhat,
               sc.cn.d.l.ss.yhat,
               sc.cn.d.l.ssiar.yhat),
  y.obs = c(sc.cn.d.l$cv$glmnet_1$cv.estimates$y.obs,
            sc.cn.d.l$cv$`ss_s0=0.27_s1=7.5_alpha=1`$cv.estimates$y.obs,
            sc.cn.d.l$cv$`ssiar_s0=0.26_s1=6_alpha=1`$cv.estimates$y.obs),
  comparison = "CN vs. Dem.")

# saveRDS(sc.cn.d.l.yhat,
#         file = paste0(figdir, "thickness_cn_d_l_yhat_grid.rds")
# )

sc.cn.d.en.glmnet.yhat <- apply(
  sc.cn.d.en$cv$glmnet_0.5$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.d.en$cv$glmnet_0.5$cv.estimates))),
  1, mean
)
sc.cn.d.en.ss.yhat <- apply(
  sc.cn.d.en$cv$`ss_s0=0.26_s1=10_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.d.en$cv$`ss_s0=0.26_s1=10_alpha=0.5`$cv.estimates))),
  1, mean
)
sc.cn.d.en.ssiar.yhat <- apply(
  sc.cn.d.en$cv$`ssiar_s0=0.28_s1=10_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.d.en$cv$`ssiar_s0=0.28_s1=10_alpha=0.5`$cv.estimates))),
  1, mean
)

sc.cn.d.en.yhat <- data.frame(
  model = c(rep("glmnet", length(sc.cn.d.en.glmnet.yhat)),
            rep("ss", length(sc.cn.d.en.ss.yhat)),
```

```
            rep("ss_iar", length(sc.cn.d.en.ssiar.yhat))),
  alpha = rep(0.5, 3 * length(sc.cn.d.en.glmnet.yhat)),
  yhat.avg = c(sc.cn.d.en.glmnet.yhat,
               sc.cn.d.en.ss.yhat,
               sc.cn.d.en.ssiar.yhat),
  y.obs = c(sc.cn.d.en$cv$glmnet_0.5$cv.estimates$y.obs,
            sc.cn.d.en$cv$`ss_s0=0.26_s1=10_alpha=0.5`$cv.estimates$y.obs,
            sc.cn.d.en$cv$`ssiar_s0=0.28_s1=10_alpha=0.5`$cv.estimates$y.obs),
  comparison = "CN vs. Dem.")

# saveRDS(sc.cn.d.en.yhat,
#         file = paste0(figdir, "thickness_cn_d_en_yhat_grid.rds")
# )

sc.cn.mci.l.glmnet.yhat <- apply(
  sc.cn.mci.l$cv$glmnet_1$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.mci.l$cv$glmnet_1$cv.estimates))),
  1, mean
)
sc.cn.mci.l.ss.yhat <- apply(
  sc.cn.mci.l$cv$`ss_s0=0.15_s1=7_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.mci.l$cv$`ss_s0=0.15_s1=7_alpha=1`$cv.estimates))),
  1, mean
)
sc.cn.mci.l.ssiar.yhat <- apply(
  sc.cn.mci.l$cv$`ssiar_s0=0.14_s1=4_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*", names(sc.cn.mci.l$cv$`ssiar_s0=0.14_s1=4_alpha=1`$cv.estimates))),
  1, mean
)

sc.cn.mci.l.yhat <- data.frame(
  model = c(rep("glmnet", length(sc.cn.mci.l.glmnet.yhat)),
            rep("ss", length(sc.cn.mci.l.ss.yhat)),
            rep("ss_iar", length(sc.cn.mci.l.ssiar.yhat))),
  alpha = rep(1, 3 * length(sc.cn.mci.l.glmnet.yhat)),
  yhat.avg = c(sc.cn.mci.l.glmnet.yhat,
               sc.cn.mci.l.ss.yhat,
               sc.cn.mci.l.ssiar.yhat),
  y.obs = c(sc.cn.mci.l$cv$glmnet_1$cv.estimates$y.obs,
            sc.cn.mci.l$cv$`ss_s0=0.15_s1=7_alpha=1`$cv.estimates$y.obs,
            sc.cn.mci.l$cv$`ssiar_s0=0.14_s1=4_alpha=1`$cv.estimates$y.obs),
  comparison = "CN vs. MCI")

# saveRDS(sc.cn.mci.l.yhat,
#         file = paste0(figdir, "thickness_cn_mci_l_yhat_grid.rds")
# )

sc.cn.mci.en.glmnet.yhat <- apply(
  sc.cn.mci.en$cv$glmnet_0.5$cv.estimates %>%
    select(grep("y.fitted.*", names(sc.cn.mci.en$cv$glmnet_0.5$cv.estimates))),
  1, mean
```

```
)
sc.cn.mci.en.ss.yhat <- apply(
  sc.cn.mci.en$cv$`ss_s0=0.14_s1=4.5_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.cn.mci.en$cv$`ss_s0=0.14_s1=4.5_alpha=0.5`$cv.estimates))),
  1, mean
)

sc.cn.mci.en.ssiar.yhat <- apply(
  sc.cn.mci.en$cv$`ssiar_s0=0.15_s1=7.5_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*", names(sc.cn.mci.en$cv$`ssiar_s0=0.15_s1=7.5_alpha=0.5`$cv.estimates))),
  1, mean
)

sc.cn.mci.en.yhat <- data.frame(
  model = c(rep("glmnet", length(sc.cn.mci.en.glmnet.yhat)),
            rep("ss", length(sc.cn.mci.en.ss.yhat)),
            rep("ss_iar", length(sc.cn.mci.en.ssiar.yhat))),
  alpha = rep(0.5, 3 * length(sc.cn.mci.en.glmnet.yhat)),
  yhat.avg = c(sc.cn.mci.en.glmnet.yhat,
               sc.cn.mci.en.ss.yhat,
               sc.cn.mci.en.ssiar.yhat),
  y.obs = c(sc.cn.mci.en$cv$glmnet_0.5$cv.estimates$y.obs,
            sc.cn.mci.en$cv$`ss_s0=0.14_s1=4.5_alpha=0.5`$cv.estimates$y.obs,
            sc.cn.mci.en$cv$`ssiar_s0=0.15_s1=7.5_alpha=0.5`$cv.estimates$y.obs),
  comparison = "CN vs. MCI")

# saveRDS(sc.cn.mci.en.yhat,
#         file = paste0(figdir, "thickness_cn_mci_en_yhat_grid.rds")
# )

sc.mci.d.l.glmnet.yhat <- apply(
  sc.mci.d.l$cv$glmnet_1$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.mci.d.l$cv$glmnet_1$cv.estimates))),
  1, mean
)
sc.mci.d.l.ss.yhat <- apply(
  sc.mci.d.l$cv$`ss_s0=0.18_s1=7.5_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.mci.d.l$cv$`ss_s0=0.18_s1=7.5_alpha=1`$cv.estimates))),
  1, mean
)
sc.mci.d.l.ssiar.yhat <- apply(
  sc.mci.d.l$cv$`ssiar_s0=0.14_s1=4_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.mci.d.l$cv$`ssiar_s0=0.14_s1=4_alpha=1`$cv.estimates))),
  1, mean
)

sc.mci.d.l.yhat <- data.frame(
  model = c(rep("glmnet", length(sc.mci.d.l.glmnet.yhat)),
            rep("ss", length(sc.mci.d.l.ss.yhat)),
```

```
            rep("ss_iar", length(sc.mci.d.l.ssiar.yhat))),
  alpha = rep(1, 3 * length(sc.mci.d.l.glmnet.yhat)),
  yhat.avg = c(sc.mci.d.l.glmnet.yhat,
               sc.mci.d.l.ss.yhat,
               sc.mci.d.l.ssiar.yhat),
  y.obs = c(sc.mci.d.l$cv$glmnet_1$cv.estimates$y.obs,
            sc.mci.d.l$cv$`ss_s0=0.18_s1=7.5_alpha=1`$cv.estimates$y.obs,
            sc.mci.d.l$cv$`ssiar_s0=0.14_s1=4_alpha=1`$cv.estimates$y.obs),
  comparison = "MCI vs. Dem.")

# saveRDS(sc.mci.d.l.yhat, file = paste0(figdir, "thickness_mci_d_l_yhat_grid.rds"))

sc.mci.d.en.glmnet.yhat <- apply(
  sc.mci.d.en$cv$glmnet_0.5$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.mci.d.en$cv$glmnet_0.5$cv.estimates))),
  1, mean
)
sc.mci.d.en.ss.yhat <- apply(
  sc.mci.d.en$cv$`ss_s0=0.14_s1=7_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.mci.d.en$cv$`ss_s0=0.14_s1=7_alpha=0.5`$cv.estimates))),
  1, mean
)

sc.mci.d.en.ssiar.yhat <- apply(
  sc.mci.d.en$cv$`ssiar_s0=0.14_s1=5.5_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(sc.mci.d.en$cv$`ssiar_s0=0.14_s1=5.5_alpha=0.5`$cv.estimates))),
  1, mean
)

sc.mci.d.en.yhat <- data.frame(
  model = c(rep("glmnet", length(sc.mci.d.en.glmnet.yhat)),
            rep("ss", length(sc.mci.d.en.ss.yhat)),
            rep("ss_iar", length(sc.mci.d.en.ssiar.yhat))),
  alpha = rep(0.5, 3 * length(sc.mci.d.en.glmnet.yhat)),
  yhat.avg = c(sc.mci.d.en.glmnet.yhat,
               sc.mci.d.en.ss.yhat,
               sc.mci.d.en.ssiar.yhat),
  y.obs = c(sc.mci.d.en$cv$glmnet_0.5$cv.estimates$y.obs,
            sc.mci.d.en$cv$`ss_s0=0.14_s1=7_alpha=0.5`$cv.estimates$y.obs,
            sc.mci.d.en$cv$`ssiar_s0=0.14_s1=5.5_alpha=0.5`$cv.estimates$y.obs),
  comparison = "MCI vs. Dem.")

# saveRDS(sc.mci.d.en.yhat, file = paste0(figdir, "thickness_mci_d_en_yhat_grid.rds"))
```

This block wrangles the data for ROC curves for Tau PET data.

```
#figdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/figures_rdata/
tau.bl.cn.d.l.glmnet.yhat <- apply(
  tau.bl.cn.d.l$cv$glmnet_1$cv.estimates %>%
    select(grep("y.fitted.*",
```

```
                names(tau.bl.cn.d.l$cv$glmnet_1$cv.estimates))),
  1, mean
)
tau.bl.cn.d.l.ss.yhat <- apply(
  tau.bl.cn.d.l$cv$`ss_s0=0.5_s1=10_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.d.l$cv$`ss_s0=0.5_s1=10_alpha=1`$cv.estimates))),
  1, mean
)
tau.bl.cn.d.l.ssiar.yhat <- apply(
  tau.bl.cn.d.l$cv$`ssiar_s0=0.37_s1=8_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.d.l$cv$`ssiar_s0=0.37_s1=8_alpha=1`$cv.estimates))),
  1, mean
)

tau.bl.cn.d.l.yhat <- data.frame(
  model = c(rep("glmnet", length(tau.bl.cn.d.l.glmnet.yhat)),
            rep("ss", length(tau.bl.cn.d.l.ss.yhat)),
            rep("ss_iar", length(tau.bl.cn.d.l.ssiar.yhat))),
  alpha = rep(1, 3 * length(tau.bl.cn.d.l.glmnet.yhat)),
  yhat.avg = c(tau.bl.cn.d.l.glmnet.yhat,
               tau.bl.cn.d.l.ss.yhat,
               tau.bl.cn.d.l.ssiar.yhat),
  y.obs = c(tau.bl.cn.d.l$cv$glmnet_1$cv.estimates$y.obs,
            tau.bl.cn.d.l$cv$`ss_s0=0.5_s1=10_alpha=1`$cv.estimates$y.obs,
            tau.bl.cn.d.l$cv$`ssiar_s0=0.37_s1=8_alpha=1`$cv.estimates$y.obs),
  comparison = "CN vs. Dem.")

#saveRDS(tau.bl.cn.d.l.yhat, file = paste0(figdir, "tau_cn_d_l_yhat_grid.rds"))

tau.bl.cn.d.en.glmnet.yhat <- apply(
  tau.bl.cn.d.en$cv$glmnet_0.5$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.d.en$cv$glmnet_0.5$cv.estimates))),
  1, mean
)
tau.bl.cn.d.en.ss.yhat <- apply(
  tau.bl.cn.d.en$cv$`ss_s0=0.27_s1=10_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.d.en$cv$`ss_s0=0.27_s1=10_alpha=0.5`$cv.estimates))),
  1, mean
)
tau.bl.cn.d.en.ssiar.yhat <- apply(
  tau.bl.cn.d.en$cv$`ssiar_s0=0.27_s1=9.5_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.d.en$cv$`ssiar_s0=0.27_s1=9.5_alpha=0.5`$cv.estimates))),
  1, mean
)

tau.bl.cn.d.en.yhat <- data.frame(
  model = c(rep("glmnet", length(tau.bl.cn.d.en.glmnet.yhat)),
            rep("ss", length(tau.bl.cn.d.en.ss.yhat)),
```

```
            rep("ss_iar", length(tau.bl.cn.d.en.ssiar.yhat))),
  alpha = rep(0.5, 3 * length(tau.bl.cn.d.en.glmnet.yhat)),
  yhat.avg = c(tau.bl.cn.d.en.glmnet.yhat,
               tau.bl.cn.d.en.ss.yhat,
               tau.bl.cn.d.en.ssiar.yhat),
  y.obs = c(tau.bl.cn.d.en$cv$glmnet_0.5$cv.estimates$y.obs,
            tau.bl.cn.d.en$cv$`ss_s0=0.27_s1=10_alpha=0.5`$cv.estimates$y.obs,
            tau.bl.cn.d.en$cv$`ssiar_s0=0.27_s1=9.5_alpha=0.5`$cv.estimates$y.obs),
  comparison = "CN vs. Dem."
)

#saveRDS(tau.bl.cn.d.en.yhat, file = paste0(figdir, "tau_cn_d_en_yhat_grid.rds"))

tau.bl.cn.mci.l.glmnet.yhat <- apply(
  tau.bl.cn.mci.l$cv$glmnet_1$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.mci.l$cv$glmnet_1$cv.estimates))),
  1, mean
)
tau.bl.cn.mci.l.ss.yhat <- apply(
  tau.bl.cn.mci.l$cv$`ss_s0=0.06_s1=7.5_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.mci.l$cv$`ss_s0=0.06_s1=7.5_alpha=1`$cv.estimates))),
  1, mean
)

tau.bl.cn.mci.l.ssiar.yhat <- apply(
  tau.bl.cn.mci.l$cv$`ssiar_s0=0.28_s1=5.5_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
               names(tau.bl.cn.mci.l$cv$`ssiar_s0=0.28_s1=5.5_alpha=1`$cv.estimates))),
  1, mean
)

tau.bl.cn.mci.l.yhat <- data.frame(
  model = c(rep("glmnet", length(tau.bl.cn.mci.l.glmnet.yhat)),
            rep("ss", length(tau.bl.cn.mci.l.ss.yhat)),
            rep("ss_iar", length(tau.bl.cn.mci.l.ssiar.yhat))),
  alpha = rep(1, 3 * length(tau.bl.cn.mci.l.glmnet.yhat)),
  yhat.avg = c(tau.bl.cn.mci.l.glmnet.yhat,
               tau.bl.cn.mci.l.ss.yhat,
               tau.bl.cn.mci.l.ssiar.yhat),
  y.obs = c(tau.bl.cn.mci.l$cv$glmnet_1$cv.estimates$y.obs,
            tau.bl.cn.mci.l$cv$`ss_s0=0.06_s1=7.5_alpha=1`$cv.estimates$y.obs,
            tau.bl.cn.mci.l$cv$`ssiar_s0=0.28_s1=5.5_alpha=1`$cv.estimates$y.obs),
  comparison = "CN vs. MCI")

#saveRDS(tau.bl.cn.mci.l.yhat, file = paste0(figdir, "tau_cn_mci_l_yhat_grid.rds"))

tau.bl.cn.mci.en.glmnet.yhat <- apply(
  tau.bl.cn.mci.en$cv$glmnet_0.5$cv.estimates %>%
    select(grep("y.fitted.*",
               names(tau.bl.cn.mci.en$cv$glmnet_0.5$cv.estimates))),
  1, mean
```

```
)
tau.bl.cn.mci.en.ss.yhat <- apply(
  tau.bl.cn.mci.en$cv$`ss_s0=0.11_s1=10_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.mci.en$cv$`ss_s0=0.11_s1=10_alpha=0.5`$cv.estimates))),
  1, mean
)
tau.bl.cn.mci.en.ssiar.yhat <- apply(
  tau.bl.cn.mci.en$cv$`ssiar_s0=0.25_s1=9_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.cn.mci.en$cv$`ssiar_s0=0.25_s1=9_alpha=0.5`$cv.estimates))),
  1, mean
)

tau.bl.cn.mci.en.yhat <- data.frame(
  model = c(rep("glmnet", length(tau.bl.cn.mci.en.glmnet.yhat)),
            rep("ss", length(tau.bl.cn.mci.en.ss.yhat)),
            rep("ss_iar", length(tau.bl.cn.mci.en.ssiar.yhat))),
  alpha = rep(0.5, 3 * length(tau.bl.cn.mci.en.glmnet.yhat)),
  yhat.avg = c(tau.bl.cn.mci.en.glmnet.yhat,
               tau.bl.cn.mci.en.ss.yhat,
               tau.bl.cn.mci.en.ssiar.yhat),
  y.obs = c(tau.bl.cn.mci.en$cv$glmnet_0.5$cv.estimates$y.obs,
            tau.bl.cn.mci.en$cv$`ss_s0=0.11_s1=10_alpha=0.5`$cv.estimates$y.obs,
            tau.bl.cn.mci.en$cv$`ssiar_s0=0.25_s1=9_alpha=0.5`$cv.estimates$y.obs),
  comparison = "CN vs. MCI")

#saveRDS(tau.bl.cn.mci.en.yhat, file = paste0(figdir, "tau_cn_mci_en_yhat_grid.rds"))

tau.bl.mci.d.l.glmnet.yhat <- apply(
  tau.bl.mci.d.l$cv$glmnet_1$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.mci.d.l$cv$glmnet_1$cv.estimates))),
  1, mean
)
tau.bl.mci.d.l.ss.yhat <- apply(
  tau.bl.mci.d.l$cv$`ss_s0=0.17_s1=10_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.mci.d.l$cv$`ss_s0=0.17_s1=10_alpha=1`$cv.estimates))),
  1, mean
)

tau.bl.mci.d.l.ssiar.yhat <- apply(
  tau.bl.mci.d.l$cv$`ssiar_s0=0.17_s1=8.5_alpha=1`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.mci.d.l$cv$`ssiar_s0=0.17_s1=8.5_alpha=1`$cv.estimates))),
  1, mean)

tau.bl.mci.d.l.yhat <- data.frame(
  model = c(rep("glmnet", length(tau.bl.mci.d.l.glmnet.yhat)),
            rep("ss", length(tau.bl.mci.d.l.ss.yhat)),
            rep("ss_iar", length(tau.bl.mci.d.l.ssiar.yhat))),
  alpha = rep(1, 3 * length(tau.bl.mci.d.l.glmnet.yhat)),
```

```
  yhat.avg = c(tau.bl.mci.d.l.glmnet.yhat,
               tau.bl.mci.d.l.ss.yhat,
               tau.bl.mci.d.l.ssiar.yhat),
  y.obs = c(tau.bl.mci.d.l$cv$glmnet_1$cv.estimates$y.obs,
            tau.bl.mci.d.l$cv$`ss_s0=0.17_s1=10_alpha=1`$cv.estimates$y.obs,
            tau.bl.mci.d.l$cv$`ssiar_s0=0.17_s1=8.5_alpha=1`$cv.estimates$y.obs),
  comparison = "MCI vs. Dem.")

#saveRDS(tau.bl.mci.d.l.yhat, file = paste0(figdir, "tau_mci_d_l_yhat_grid.rds"))

tau.bl.mci.d.en.glmnet.yhat <- apply(
  tau.bl.mci.d.en$cv$glmnet_0.5$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.mci.d.en$cv$glmnet_0.5$cv.estimates))),
  1, mean
)
tau.bl.mci.d.en.ss.yhat <- apply(
  tau.bl.mci.d.en$cv$`ss_s0=0.17_s1=10_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.mci.d.en$cv$`ss_s0=0.17_s1=10_alpha=0.5`$cv.estimates))),
  1, mean
)
tau.bl.mci.d.en.ssiar.yhat <- apply(
  tau.bl.mci.d.en$cv$`ssiar_s0=0.17_s1=10_alpha=0.5`$cv.estimates %>%
    select(grep("y.fitted.*",
                names(tau.bl.mci.d.en$cv$`ssiar_s0=0.17_s1=10_alpha=0.5`$cv.estimates))),
  1, mean
)

tau.bl.mci.d.en.yhat <- data.frame(
  model = c(rep("glmnet", length(tau.bl.mci.d.en.glmnet.yhat)),
            rep("ss", length(tau.bl.mci.d.en.ss.yhat)),
            rep("ss_iar", length(tau.bl.mci.d.en.ssiar.yhat))),
  alpha = rep(0.5, 3 * length(tau.bl.mci.d.en.glmnet.yhat)),
  yhat.avg = c(tau.bl.mci.d.en.glmnet.yhat,
               tau.bl.mci.d.en.ss.yhat,
               tau.bl.mci.d.en.ssiar.yhat),
  y.obs = c(tau.bl.mci.d.en$cv$glmnet_0.5$cv.estimates$y.obs,
            tau.bl.mci.d.en$cv$`ss_s0=0.17_s1=10_alpha=0.5`$cv.estimates$y.obs,
            tau.bl.mci.d.en$cv$`ssiar_s0=0.17_s1=10_alpha=0.5`$cv.estimates$y.obs),
  comparison = "MCI vs. Dem.")

#saveRDS(tau.bl.mci.d.en.yhat, file = paste0(figdir, "tau_mci_d_en_yhat_grid.rds"))
```

5.1.2 The ROC Curves Themselves

Assuming you've saved the files using the code in the previous section, you can run this chunk to put them all together and add some labels.

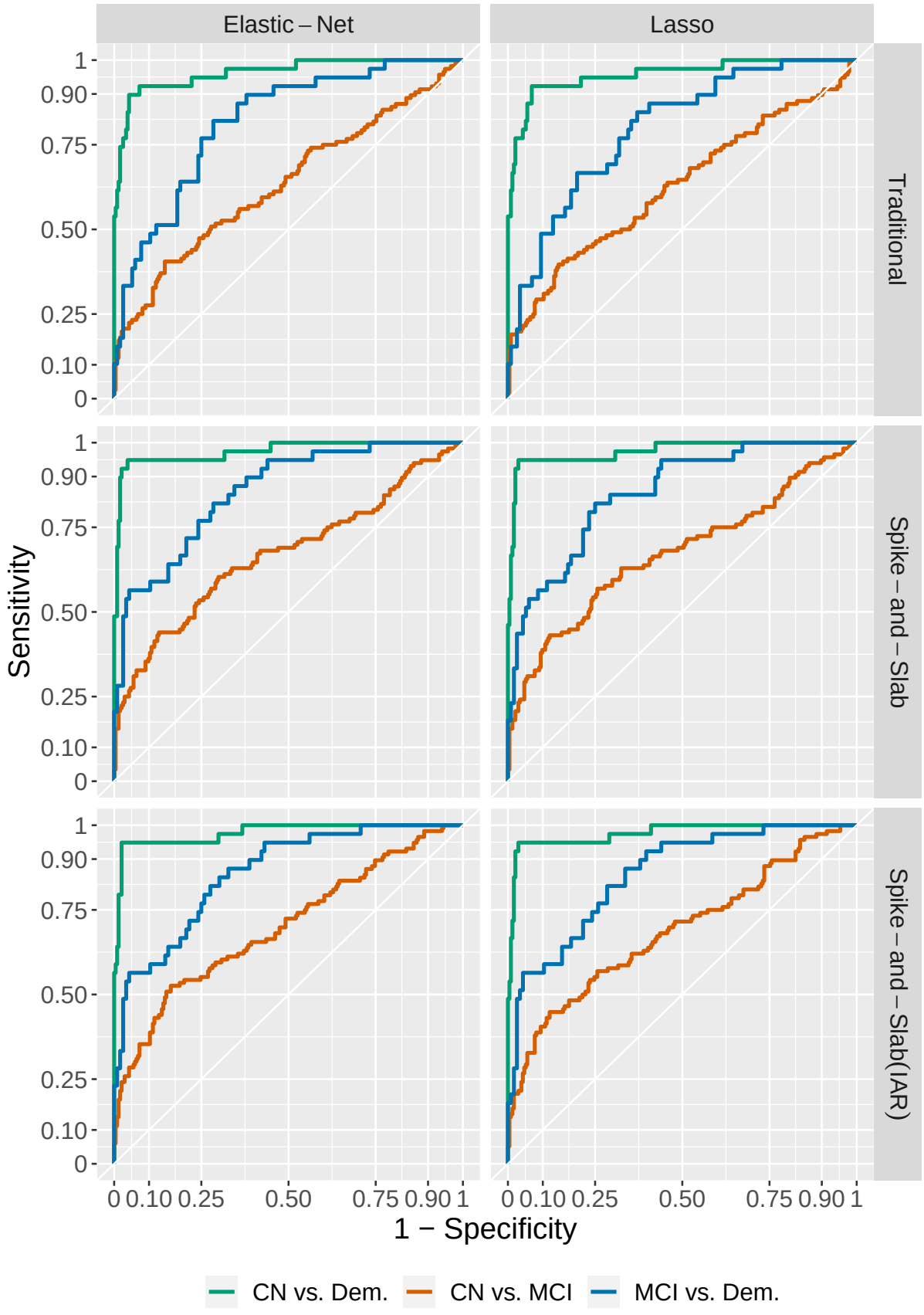
This is Figure 3 from the paper (i.e., ROC curves for cortical thickness as features).

```
ggplot(data = all.sc.yhat, aes(d = y.obs, m = yhat.avg, color = comparison)) +
  geom_roc(labels = FALSE, n.cuts = 0) +
```

```
  # geom_roc(labels = TRUE, cutoffs.at = c(0.25, 0.5, 0.75),
  #          labelround = 2, labelsize = 3) +
  style_roc(theme = theme_grey,
            ylab = "Sensitivity", xlab = "1 - Specificity") +
  scale_y_continuous(name = "Sensitivity",
                     breaks = c(0, 0.1, 0.25, 0.5, 0.75, 0.9, 1),
                     labels = c("0", "0.10", "0.25", "0.50", "0.75", "0.90", "1")) +
  scale_x_continuous(name = "1 - Specificity",
                     breaks = c(0, 0.1, 0.25, 0.5, 0.75, 0.9, 1),
                     labels = c("0", "0.10", "0.25", "0.50", "0.75", "0.90", "1")) +
  facet_grid(model.lab~alpha.lab2, labeller = labeller(alpha.lab2 = label_parsed,
                                                       model.lab = label_parsed)) +
  scale_color_manual(values = cbp.b[c(4, 7, 6)]) +
  theme(plot.title = element_text(hjust = 0.5),
        text = element_text(size = 16),
        legend.position = "bottom",
        legend.title = element_blank())
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

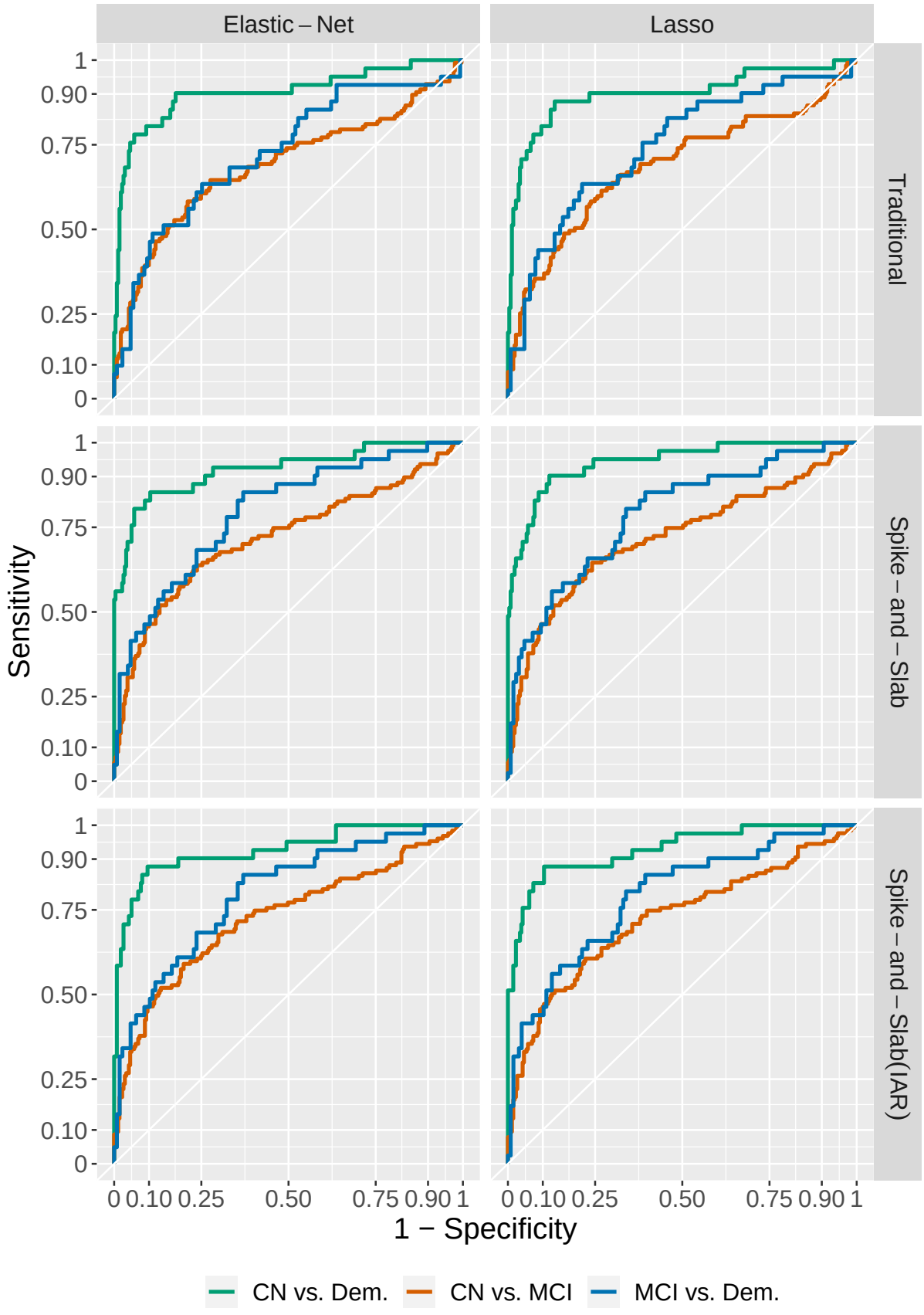

Elastic – Net
Lasso
Traditional
Spike – and – Slab
Spike – and – Slab(IAR)
Sensitivity
1 − Specificity
0
0.10
0.25
0.50
0.75
0.90
1
CN vs. Dem.
CN vs. MCI
MCI vs. Dem.

This is Figure 4 from the paper (i.e., ROC curves for tau PET used as features).

```
ggplot(data = all.tau.bl.yhat, aes(d = y.obs, m = yhat.avg, color = comparison)) +
  geom_roc(labels = FALSE, n.cuts = 0) +
  # geom_roc(labels = TRUE, cutoffs.at = c(0.25, 0.5, 0.75),
  #          labelround = 2, labelsize = 3) +
  style_roc(theme = theme_grey,
            ylab = "Sensitivity", xlab = "1 - Specificity") +
  scale_y_continuous(name = "Sensitivity",
                     breaks = c(0, 0.1, 0.25, 0.5, 0.75, 0.9, 1),
                     labels = c("0", "0.10", "0.25", "0.50", "0.75", "0.90", "1")) +
  scale_x_continuous(name = "1 - Specificity",
                     breaks = c(0, 0.1, 0.25, 0.5, 0.75, 0.9, 1),
                     labels = c("0", "0.10", "0.25", "0.50", "0.75", "0.90", "1")) +
  facet_grid(model.lab~alpha.lab2, labeller = labeller(alpha.lab2 = label_parsed,
                                                       model.lab = label_parsed)) +
  scale_color_manual(values = cbp.b[c(4, 7, 6)]) +
  theme(plot.title = element_text(hjust = 0.5),
        text = element_text(size = 16),
        legend.position = "bottom",
        legend.title = element_blank())
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```


Elastic – Net
Lasso
Traditional
Spike – and – Slab
Spike – and – Slab(IAR)
Sensitivity
1 − Specificity
0
0.10
0.25
0.50
0.75
0.90
1
CN vs. Dem.
CN vs. MCI
MCI vs. Dem.

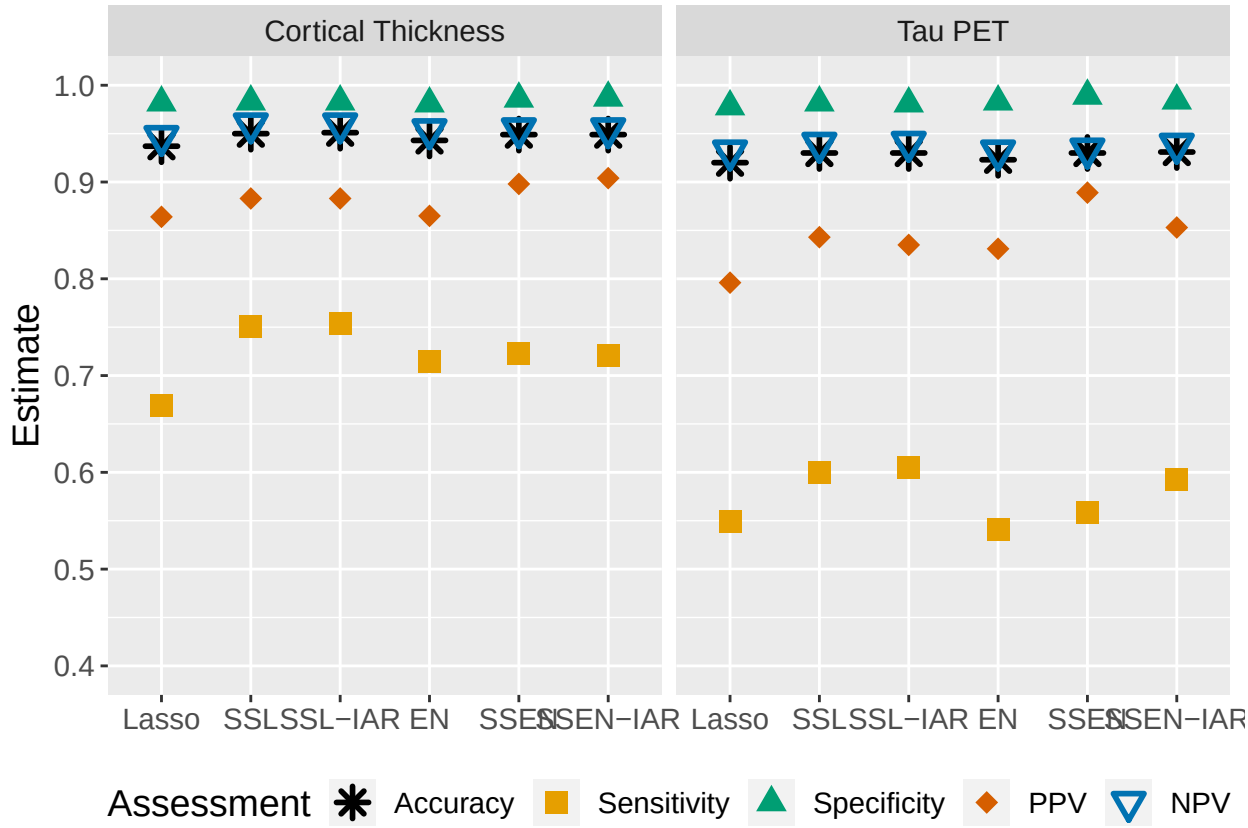
5.2 Figures for Classification Performance

We have a little bit of wrangling before plots.

```
resdir <- "C:/Users/Justin/Documents/BST/Dissertation_in_Latex/for-publishing/Rcode-ADNI/results_analys
# colorblind palettes
# colors: 1 grey/black 2 orange 3 light blue 4 green 5 yellow 6 dark blue 7 red-orange 8 pink
# The palette with grey:
cbp.g <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# The palette with black:
cbp.b <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# load results summaries
ct.smry <- readRDS(file = paste0(resdir, "ct_smry.rds"))
ct.smry.ca <- ct.smry %>%
  select(outcome, model, alpha, s0, s1, accuracy, sensitivity, specificity, ppv, npv)
tau.smry <- readRDS(file = paste0(resdir, "tau_smry.rds"))
tau.smry.ca <- tau.smry %>%
  select(outcome, model, alpha, s0, s1, accuracy, sensitivity, specificity, ppv, npv)

# Model names for figures
model <- rep(c("Lasso", "SSL", "SSL-IAR",
               "EN", "SSEN", "SSEN-IAR"),
             3)

# wrangle for figures
accuracy.wide <- cbind(Model = model,
                       Modality = c(rep("Cortical Thickness", 18), rep("Tau PET", 18)) ,
                       rbind(ct.smry.ca, tau.smry.ca)[, c(1, 6:10)]
)
all.long <- cbind(accuracy.wide[ , c(1:3)],
                  Estimate = c(accuracy.wide$accuracy,
                               accuracy.wide$sensitivity,
                               accuracy.wide$specificity,
                               accuracy.wide$ppv,
                               accuracy.wide$npv),
                  Assessment = c(rep("Accuracy", 36),
                                 rep("Sensitivity", 36),
                                 rep("Specificity", 36),
                                 rep("PPV", 36),
                                 rep("NPV", 36)))
names(all.long) <- c("Model", "Modality", "Comparison", "Estimate", "Assessment")
all.long$Model <- factor(all.long$Model,
                         levels = c("Lasso", "SSL", "SSL-IAR",
                                    "EN", "SSEN", "SSEN-IAR"))
all.long$Assessment <- factor(all.long$Assessment,
                              levels = c("Accuracy", "Sensitivity", "Specificity", "PPV", "NPV"))

# I think this `set.seed()` argument was here by accident.
# I can't find a use for it, but left it commented out just in case.
# set.seed(5845549)
```

This is Figure 5 from the paper (i.e., performance for classifying cognitive normal vs. dementia subjects).

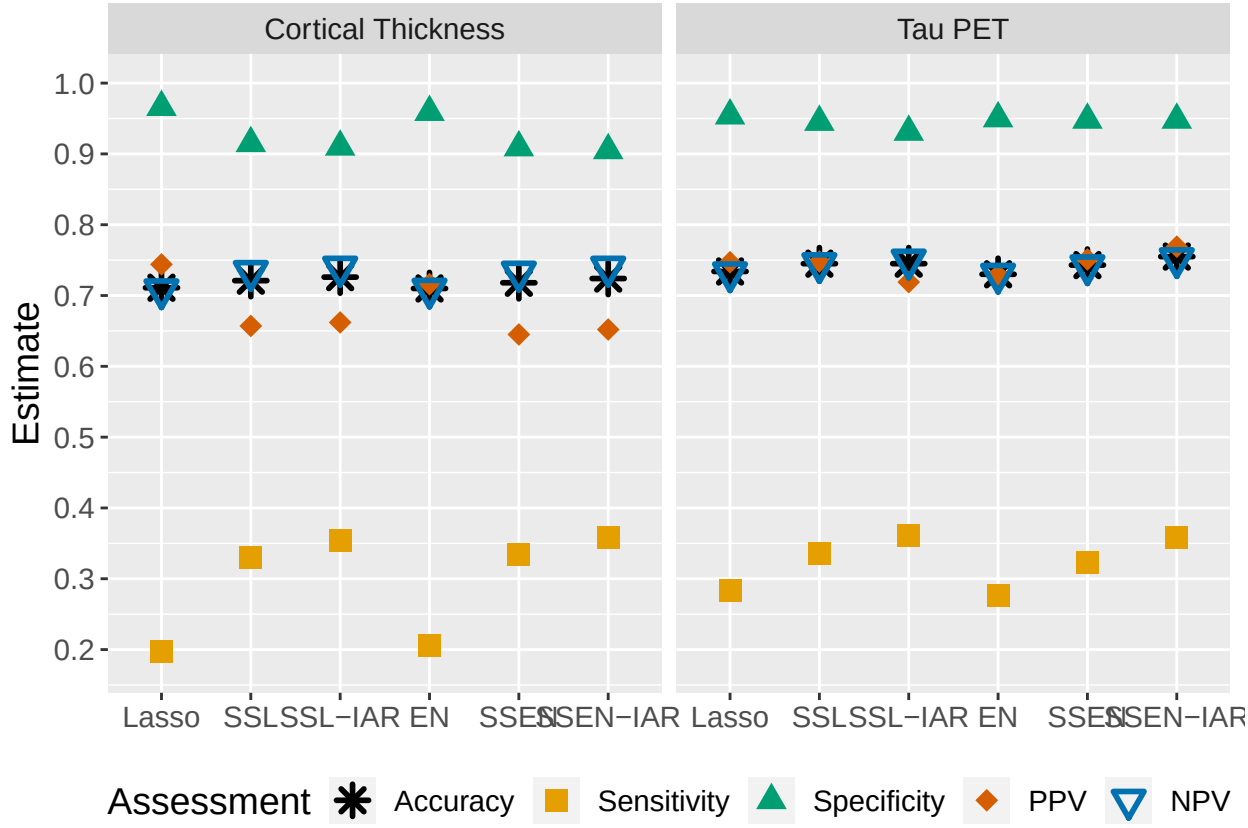
```r
# CN vs. D
cnd.long <- all.long %>%
  filter(Comparison == "CN vs. Dem.")

ggplot(data = cnd.long,
       mapping = aes(y = Estimate,
                     x = Model,
                     colour = Assessment,
                     shape = Assessment)) +
  geom_point(size = 3, stroke = 1.5) +
  scale_shape_manual(values = c(8, 15, 17, 18, 25)) +
  scale_color_manual(values = cbp.b[c(1, 2, 4, 7, 6)]) +
  # scale_fill_manual(values = cbp.b[c(1, 2, 4, 7, 3 )]) +
  facet_wrap(cnd.long$Modality) +
  scale_x_discrete(name = NULL) +
  scale_y_continuous(limits = c(0.4, 1),
                     breaks = seq(0.4, 1, 0.1)) +
  theme(plot.title = element_text(hjust = 0.5),
        text = element_text(size = 14),
        legend.position = "bottom")
```



This is Figure 6 from the paper (i.e., performance for classifying cognitive normal vs. MCI subjects).

```r
# CN vs. MCI
cnmci.long <- all.long %>%
  filter(Comparison == "CN vs. MCI")
```

```r
ggplot(data = cnmci.long,
       mapping = aes(y = Estimate,
                     x = Model,
                     colour = Assessment,
                     shape = Assessment)) +
  geom_point(size = 3, stroke = 1.5) +
  scale_shape_manual(values = c(8, 15, 17, 18, 25)) +
  scale_color_manual(values = cbp.b[c(1, 2, 4, 7, 6)]) +
  # scale_fill_manual(values = cbp.b[c(1, 2, 4, 7, 3 )]) +
  facet_wrap(cnd.long$Modality) +
  scale_x_discrete(name = NULL) +
  scale_y_continuous(limits = c(0.18, 1),
                     breaks = seq(0.2, 1, 0.1)) +
  theme(plot.title = element_text(hjust = 0.5),
        text = element_text(size = 14),
        legend.position = "bottom")
```



This is Figure 7 from the paper (i.e., performance for classifying MCI vs. dementia subjects).

```r
# MCI vs. D
mcid.long <- all.long %>%
  filter(Comparison == "MCI vs. Dem.")
ggplot(data = mcid.long,
       mapping = aes(y = Estimate,
                     x = Model,
                     colour = Assessment,
```

```
                    shape = Assessment)) +
geom_point(size = 3, stroke = 1.5) +
scale_shape_manual(values = c(8, 15, 17, 18, 25)) +
scale_color_manual(values = cbp.b[c(1, 2, 4, 7, 6)]) +
# scale_fill_manual(values = cbp.b[c(1, 2, 4, 7, 3 )]) +
facet_wrap(cnd.long$Modality) +
scale_x_discrete(name = NULL) +
scale_y_continuous(limits = c(0.15, 1),
                   breaks = seq(0.2, 1, 0.1)) +
theme(plot.title = element_text(hjust = 0.5),
      text = element_text(size = 14),
      legend.position = "bottom")
```