Data Supplement for Squeglia et al., Neural Predictors of Initiating Alcohol Use During Adolescence. Am J Psychiatry (doi:10.1176/appi.ajp.2016.15121587)

SUPPLEMENTARY MATERIAL

To better understand the unique contributions of each class of variables, models were run on 26 neuropsychological and 136 neuroimaging variables separately.

Neuropsychological performance only model. For the variable set that only included neuropsychological test variables (26 variables; see Table 2), 8 variables were identified as important: CVLT list A total 1 to 5 raw total, D-KEFS Trails Number-Letter Switching time to complete (sec), WASI Vocab raw, WASI Similarities raw, WRAT-3 Reading raw, Digit Vigilance total time to complete (sec), WASI Block Design raw, and WASI Matrix Reasoning raw.

Neuroimaging only model. For the variable set that only included neuroimaging variables (cortical thickness and BOLD response for each of the 68 brain regions), 21 variables were identified as important including 15 cortical thickness regions (left banks of the superior temporal sulcus, left lateral occipital, left lingual, left rostral anterior cingulate, left superior parietal, left supramarginal, left temporal pole, left transverse temporal, right middle temporal, right pars orbitalis, right precuneus, right superior frontal, right superior parietal, right frontal pole, and right temporal pole) and 6 BOLD regions (left middle temporal, left precuneus, right caudal anterior cingulate, right posterior cingulate, right precuneus, and right superior temporal; see Supplementary Table 1).

Supplementary Random Forests R Scripts to be Used for Replication.

Feature Selection Script

```
### This script runs the Random Forests (RF) feature selection analysis pathway as developed by Tali Ball, PhD
### This script is recommended for samples of less than 200 subjects
### This was used in Squeglia, Ball, et al (2016) American Journal of Psychiatry
###

############ 5 Parameters to be changed by the user: #################
# (1) Where your inputs are and your outputs will be
workingDirectory <- "~/Documents/RandomForest/"
# (2) csv file with all your data, including a header (the first row is the variable names)
inputFile <- "FINAL_Dataset.csv"
# (3) column number in csv file containing your outcome variable
outcomeColumn <- 2
# (4) column numbers in csv file containing your desired predictor variables
predictorColumns <- c(29:54)
# (5) prefix for output files
outputPrefix <- "randomforestoutput"
#######################################################################
#Setup
if (!"randomForest" %in% installed.packages()[,"Package"]) install.packages("randomForest")
require(randomForest)
setwd(workingDirectory)

mydata <- read.csv(inputFile, header=T)
set.seed(33)
outcomeName <- names(mydata)[outcomeColumn]
mydata <- mydata[,c(outcomeColumn,predictorColumns)]

#Handle missing data with rfimpute
f <- paste(outcomeName, "~ .")
mydata.imputed <-rfImpute(eval(parse(text = f)),data=mydata)
outcome <- mydata[,which(names(mydata)==outcomeName)]
predictors <- mydata.imputed[,-which(names(mydata)==outcomeName)]
mydataframe <- as.data.frame(cbind(outcome,predictors))

##### Step 1: Run 500 RFs to find median importance ratings for each predictor ######

repnum <- 500

# Makes empty output arrays
OOBErrArray <- array(0,repnum)
PermutImpArray <- array(0,c(dim(mydataframe)[2]-1,repnum))
```

```
# Loops through many RFs, store data in output arrays that were just made
for (i in 1: repnum) {
    myRF <- randomForest(formula = outcome ~ ., data = mydataframe, ntree = 2000,
importance = T, proximity = T)
    OOBErrArray[i] <- myRF$err.rate[dim(myRF$err.rate)[1],1]
    PermutImpArray[,i] <- myRF$importance[,3]
}

# Computes median importance and tacks it onto the arrays
MedianPermutImp <- array(0,dim(PermutImpArray)[1])
for (i in 1: dim(PermutImpArray)[1]) {
    MedianPermutImp[i] <- median(PermutImpArray[i,])
}
# The finished product will have a row for each predictor and a column for each repetition of the
RF that you did above, plus one column at the end that computes the median
PermutImpArray <- cbind(PermutImpArray,MedianPermutImp)

###### Step 2: Figure out which variables are the most important ######

# Set the number of predictors you'll have in your final model - currently using all variables with
median importance bigger than the absolute value of the most negative value
  prednum <- length(which(MedianPermutImp>abs(min(MedianPermutImp))))
  prednumMinus1 <- prednum-1
  n <- length(MedianPermutImp)
  reducedColNums <- which(MedianPermutImp>=sort(MedianPermutImp,partial=n-
prednumMinus1)[n-prednumMinus1])+1

###### Save outputs ######
PermutImpArrayName <- paste0(outputPrefix,"_PermutImpArray")
save(PermutImpArray,file=PermutImpArrayName)
OOBErrArrayName <- paste0(outputPrefix,"_OOBErrArray")
save(OOBErrArray,file=OOBErrArrayName)
```

Cross-Validation Script

```
### This script runs a Random Forests (RF) prediction analysis pathway using k-fold cross-
validation developed by Tali Ball, PhD
### This script is recommended for samples of greater than 200 subjects (not used in Squeglia,
Ball, et al (2016) American Journal of Psychiatry because of sample size)

############ 6 Parameters to be changed by the user: #################
# (1) Where your inputs are and your outputs will be
workingDirectory <- "~/Documents/RandomForest/"
# (2) csv file with all your data, including a header (the first row is the variable names)
```

```
inputFile <- "FINAL_Dataset.csv"
# (3) column number in csv file containing your outcome variable
outcomeColumn <- 2
# (4) column numbers in csv file containing your desired predictor variables
predictorColumns <- c(29:54)
# (5) number of folds
nfold <- 5
# (6) prefix for output files
outputPrefix <- "randomforestoutput_5fold"
######################################################################
#Setup
if (!"randomForest" %in% installed.packages()[,"Package"]) install.packages("randomForest")
if (!"TunePareto" %in% installed.packages()[,"Package"]) install.packages("TunePareto")
require(randomForest)
require(TunePareto)
setwd(workingDirectory)

mydata <- read.csv(inputFile, header=T)
set.seed(33)
outcomeName <- names(mydata)[outcomeColumn]
mydata <- mydata[,c(outcomeColumn,predictorColumns)]

OOBErrArrayList <- list()
PermutImpArrayList <- list()
SelectedVars <- list()
Accuracy <- array(0,nfold)
Confusion <- data.frame(matrix(ncol=4,nrow=nfold))
AllRFObjs <- list()

#Handle missing data with rfimpute
f <- paste(outcomeName, "~ .")
mydata.imputed <-rfImpute(eval(parse(text = f)),data=mydata)
outcome <- mydata[,which(names(mydata)==outcomeName)]
predictors <- mydata.imputed[,-which(names(mydata)==outcomeName)]
mydataframe <- as.data.frame(cbind(outcome,predictors))

##### Set up the folds/cross validation
whichfold <- as.numeric()
# Generates nfold sets of row numbers
foldlist <- generateCVRuns(outcome,ntimes=1,nfold=nfold, leaveOneOut=F, stratified=T)
# Assigns each row the fold for which it will be the test set
for (fold in 1 : nfold) {
  whichfold[foldlist[[1]][fold][[1]]] <- fold
}

for (fold in 1: nfold) {
```

```
  cat(sprintf("Fold # %g / %g \n", fold, nfold))
  # In each fold, the rows assigned to that number are the test set, the rest are the training set
  trainingdata <- mydataframe[whichfold!=fold,]
  testdata <- mydataframe[whichfold==fold,]

####### Step 1: Run 500 RFs to find median importance ratings for each predictor using the
training set #######

repnum <- 500

# Makes empty output arrays
OOBErrArray <- array(0,repnum)
PermutImpArray <- array(0,c(dim(trainingdata)[2]-1,repnum))

# Loops through many RFs, store data in output arrays that were just made
for (i in 1: repnum) {
     myRF <- randomForest(formula = outcome ~ ., data = trainingdata, ntree = 2000, importance
= T, proximity = T)
     OOBErrArray[i] <- myRF$err.rate[dim(myRF$err.rate)[1],1]
     PermutImpArray[,i] <- myRF$importance[,3]
}

# Computes median importance and tacks it onto the arrays
MedianPermutImp <- array(0,dim(PermutImpArray)[1])
for (i in 1: dim(PermutImpArray)[1]) {
     MedianPermutImp[i] <- median(PermutImpArray[i,])
}
# The finished product will have a row for each predictor and a column for each repetition of the
RF that you did above, plus one column at the end that computes the median
PermutImpArray <- cbind(PermutImpArray,MedianPermutImp)

# Store the outputs of this process for this fold
  OOBErrArrayList[[fold]] <- OOBErrArray
  PermutImpArrayList[[fold]] <- PermutImpArray

###### Step 2: Figure out which variables you will keep in your final RF ######

# Set the number of predictors you'll have in your final model - currently using all variables with
median importance bigger than the absolute value of the most negative value
  prednum <- length(which(MedianPermutImp>abs(min(MedianPermutImp))))
  prednumMinus1 <- prednum-1
  n <- length(MedianPermutImp)
  reducedColNums <- which(MedianPermutImp>=sort(MedianPermutImp,partial=n-
prednumMinus1)[n-prednumMinus1])+1
  SelectedVars[[fold]] <- names(mydataframe)[reducedColNums]
```

```
####### Step 3: Run RF with just the variables you kept in step 2, using the test set #######

mydataframe_reduced <- testdata[,c(1,reducedColNums)]
myRF_reduced <- randomForest(formula=outcome ~ ., data=mydataframe_reduced, ntree=2000,
importance=T, proximity=T)

  Accuracy[fold] <- 1-myRF_reduced$err.rate[dim(myRF_reduced$err.rate)[1],1]
  Confusion[fold,] <-
c(myRF_reduced$confusion[1,1],myRF_reduced$confusion[1,2],myRF_reduced$confusion[2,1]
,myRF_reduced$confusion[2,2])

  AllRFObjs[[fold]] <- myRF_reduced

}

colnames(Confusion) <- c("TruePos","FalseNeg","FalsePos","TrueNeg")


### Save the outputs
PermutImpArrayName <- paste0(outputPrefix,"_PermutImpArray")
save(PermutImpArrayList,file=PermutImpArrayName)
OOBErrArrayName <- paste0(outputPrefix,"_OOBErrArray")
save(OOBErrArrayList,file=OOBErrArrayName)
RFObjectName <- paste0(outputPrefix,"_RFObject")
save(AllRFObjs,file=RFObjectName)
AccuracyName <- paste0(outputPrefix,"_Accuracy")
save(Accuracy,file=AccuracyName)
ConfusionName <- paste0(outputPrefix,"_ConfusionMat")
save(Confusion,file=ConfusionName)
```

Supplementary Table 1. Comparison of Model 3 and Neuroimaging variables only in 137 adolescents (67 Controls, 70 Moderate-Heavy Alcohol Initiators). Model 3 included all of the Variables from Models 1 + 2 + Neuroimaging variables [cortical thickness and blood oxygen level dependent (BOLD) response during a visual working memory task. Desikan (1) brain region location is specified using R=right hemisphere, L=left hemisphere, as well as the neuroimaging index CT=cortical thickness and BOLD=BOLD response contrast during a visual working memory task (6-dot supra-span relative to the 2-dot sub-span condition). The variables that differed between Model 3 and Neuroimaging Only model were: right rostral middle frontal cortical thickness, right frontal pole BOLD, and left temporal pole cortical thickness.

CORTICAL THICKNESS AND BOLD REGIONS	Model 3	Neuroimaging Only
Based on Desikan atlas (1); 34 regions per hemisphere, cortical thickness and fMRI measures for each region listed; 34 x 2 x 2=136 total variables		
1. Banks of superior temporal sulcus	L-CT	L-CT
2. Caudal anterior cingulate	R-BOLD	R-BOLD
3. Caudal middle frontal		
4. Cuneus		
5. Entorhinal		
6. Fusiform		
7. Inferior parietal		
8. Inferior temporal		
9. Isthmus cingulate		
10. Lateral occipital	L-CT	L-CT
11. Lateral orbitofrontal		
12. Lingual	L-CT	L-CT
13. Medial orbitofrontal		
14. Middle temporal	R-CT, L-BOLD	R-CT, L-BOLD
15. Parahippocampal		
16. Paracentral		
17. Pars opercularis		
18. Pars orbitalis	R-CT	R-CT
19. Pars triangularis		
20. Pericalcarine		
21. Postcentral		
22. Posterior cingulate	R-BOLD	R-BOLD
23. Precentral		
24. Precuneus	R-CT, L-BOLD, R-BOLD	R-CT, L-BOLD, R-BOLD
25. Rostral anterior cingulate	L-CT	L-CT
26. Rostral middle frontal	R-CT	
27. Superior frontal	R-CT	R-CT
28. Superior parietal	L-CT,	L-CT,

	R-CT	R-CT
29. Superior temporal	R-BOLD	R-BOLD
30. Supramarginal	L-CT	L-CT
31. Frontal pole	R-CT, R-BOLD	R-CT
32. Temporal pole	R-CT	R-CT, L-CT
33. Transverse temporal	L-CT	L-CT
34. Insula		

Supplemental References for Neuropsychological Testing Materials:

1. Delis DC, Kaplan E, Kramer JH: The Delis-Kaplan Executive Function System: Examiner's Manual. in The Psychological Corporation. San Antonio2001.
2. Delis DC, Kramer JH, Kaplan E, Ober BA: Manual for the California Verbal Learning Test–Children's Version. San Antonio, TX, The Psychological Corporation; 1994.
3. Delis DC, Kramer JH, Kaplan E, Ober BA: The California Verbal Learning Test–Second Edition. San Antonio, TX, The Psychological Corporation.; 2000.
4. Lewis RF: Digit Vigilance Test. Odessa, FL, Psychological Assessment Resources; 1995.
5. Rey A, Osterrieth PA. Translations of excerpts from Andre Rey's "Psychological examination of traumatic encephalopathy" and P.A. Osterrieth's "The complex figure copy test" (J. Corwin & F. W. Bylsma, Trans.). The Clinical Neuropsychologist. 1993;7:3-21.
6. Wechsler D: Wechsler Intelligence Scale for Children (3rd ed.). New York, Psychological Corporation; 1991.
7. Wechsler D: Wechsler Adult Intelligence Scale (3rd ed.). San Antonio, TX, The Psychological Corporation; 1997.
8. Wechsler D: Wechsler Abbreviated Scale of Intelligence. San Antonio, TX, The Psychological Corporation; 1999.
9. Wechsler D: Wechsler Adult Intelligence Scale–Fourth Edition. San Antonio, TX, Pearson; 2008.
10. Wilkinson GS: WRAT-3: Wide Range Achievement Test administration manual (3rd ed.). Wilmington, DE, Western Psychological Services; 1993.