```
# Date Completed: 09-05-2018
#-------------------------------------------------------------
#-------------------------------------------------------------
# Input files
#-------------------------------------------------------------
#1. A master dataset that includes the genetic data for all patients,
#   including a column indicicating training vs validation cohort.
#2. A character vector of the variants to be modelled
#   in the current iteration.
#-------------------------------------------------------------
# Output files
#-------------------------------------------------------------
# 1. .Rdata file that contains Random Forest object and
#    variable importance data for the current seed and iteration
#-------------------------------------------------------------
# Purpose
#-------------------------------------------------------------
# Build a random forest model for a single unique seed and set of
# genetic variants
#-------------------------------------------------------------

#load packages
packages <- c("batch", "tibble", "ranger", "dplyr")
invisible(lapply(packages, library, character.only = TRUE))

#assign iteration (iter) and seed (n.seed) from command line for running on H
PC system
iter=NULL; n.seed=NULL;
parseCommandArgs();

#set today's date--for saving/exporting purposes
tdate<-format(Sys.Date(),"%m-%d-%Y")

#build random forest model using ONLY training cohort data
set.seed(n.seed)
rf.model<-ranger(msdss_last~., data = training, splitrule="variance", num.tre
es = 10000, importance="permutation")

#set up file for results
results<-data.frame(imp = importance(df.range)) %>%



            rownames_to_column(var = "variant")    %>%
            setNames(c("variant",paste0("importance",n.seed)))

#save ranger object and variable importance for current seed and iteration
save(results, rf.model, file = paste0("./tmp/ranger_out_",n.seed,"_",iter,"_r
esults.Rdata"))
```

```
 # Date Completed: 09-05-2018
#-------------------------------------------------------------
#-------------------------------------------------------------#
Input files
#-------------------------------------------------------------#
1. A vector that includes sample ids for the training cohort#
2. A vector that includes sample ids for the validation
#    cohort
# 3. A character vector of the variants to be modelled in the
#    current iteration.
#-------------------------------------------------------------#
Output files
#-------------------------------------------------------------#
1. Data table including average OOB error, RMSE, and


#    correlation coefficients for the training and validation cohorts.
#-------------------------------------------------------------
# Purpose
#-------------------------------------------------------------
# Combine the model results from ALL unique seeds and generate model
# statistics for the current iteration
#-------------------------------------------------------------

#Load packages
packages <- c("batch", "matrixStats", "tibble", "ranger", "dplyr")
invisible(lapply(packages, library, character.only = TRUE))

#Today's date
tdate<-format(Sys.time(),"%m-%d-%Y")

#assign iteration (iter) from command line for running on HPC system
iter<-NULL;
parseCommandArgs()

#Empty dataframe to store variable importance for all variants in the current
interation.
merged.results<-data.frame(variant = variant_names)

#Pattern for locating .Rdata files that contain Random Forest object and vari
able importance
pat<-paste0("ranger_out_[0-9]+_",iter,"_results.Rdata")

#dir.files contains the file names for all .Rdata files correpsonding to the
current iteration
dir.files<-list.files(path = "./tmp", pattern = pat,full.names=T)

#For each seed, open .Rdata file and add variable importance correpsonding to
seed to dataframe
```

```r
for (i in dir.files){
  load(i)
  merged.results <- merged.results %>% merge(results,by="variant")
}

#Dataset that contain variable importance data only.
merged.results.VI <- merged.results %>% dplyr::select(variant,contains("impor
tance"))

#Calculate the mean VI for each variant
merged.results.VI <- merged.results.VI %>%
                        mutate(means = rowMeans(dplyr::select_if(merged.resul
ts.VI,is.numeric))) %>%
                        arrange(desc(means))

#Remove variant 1 at a time
imp_vars <- merged.results.VI$variant
imp_vars <- imp_vars[-length(imp_vars)]

#Save NEW list of variants for next iteration. New list has removed the least
important variable.
write.table(imp_vars,file=paste0("iter_",iter+1,"_variants_",tdate,".txt"),ro
w.names=F, quote=F,sep="\t",col.names=F)


#------------------------------------------------------------
# Model Statistics
#------------------------------------------------------------

#Empty vector to save the OOB error for each model
errors<-numeric()

#Empty datasets to store training/validation predictions
df.train <- data.frame(id = training_samples)
df.valid <- data.frame(id = validation_samples)


#Aggregate predictions for all seeds
for (i in dir.files){
        load(i)

        #Store error of model
        errors<-c(errors, rf.model$prediction.error)

        # OOB predictions to training data
        p.train<-data.frame(rf.model$predictions)
        df.train<-df.train %>% bind_cols(p.train)

        #Add model predictions to validation data
```

```
        p.valid<-data.frame(predict(rf.model,validation)$predictions)
        df.valid<-df.valid %>% bind_cols(p.valid)
      }

#Predictions
df.train <- df.train %>% dplyr::select_if(is.numeric) %>% mutate(model_outcom
e = rowMeans(.,na.rm=T))
df.valid <- df.valid %>% dplyr::select_if(is.numeric) %>% mutate(model_outcom
e = rowMeans(.,na.rm=T))

#Correlation statistics for training and validation cohorts
r_test_t <- cor.test(training$msdss_last,df.train$model_outcome)
r_test_v <- cor.test(validation$msdss_last,df.valid$model_outcome)

r_pvalue_t <- r_test_t$p.value
r_pvalue_v <- r_test_v$p.value

r_est_t<-r_test_t$estimate
r_est_v<-r_test_v$estimate

#RMSE
resid_t <- training$msdss_last - df.train$model_outcome
rmse_t <- sqrt(mean((resid_t)^2))

resid_v <- validation$msdss_last - df.valid$model_outcome
rmse_v <- sqrt(mean((resid_v)^2))

#Average OOB error
err<-mean(errors,na.rm=T)

df<-data.frame(iter,round(err,3),
               round(rmse_t,3), round(r_est_t,3), r_pvalue_t,
               round(rmse_v,3), round(r_est_v,3), r_pvalue_v, fix.empty.names=
F)

#Output Data
sink(file=paste0("output_",tdate,"_.txt"),append=T, type="output")
print(df)
sink()
```