Appendix S1. Related computerized programs for statistical analysis with R

For intraclass correlation coefficient calculation
T12 <- cbind(T1, T2); dim(T12)
library(psych)
t = 2
icc <- c(1: y)
for (i in 1: y) { icc[i] <- ICC(T12[, c(i, i + y)]$results$ICC[t]}

For logistic regression
library('MASS')
glm <- glm(Y ~ age…, family = binomial(link = logit), data = df)
logit.step <- step(glm, direction = 'backward')

For least absolute shrinkage and selection operator regression and 10-fold cross-validation
library(ncvreg)
X <- as.matrix(df[, 1: 613])
y <- df$pathologicaltype
n <- dim(X)[1]
p <- dim(X)[2]
cv.out <- cv.ncvreg(X, y, family = 'binomial', penalty = 'MCP', nf
olds = 10)
plot(cv.out, log.l = FALSE)

For receiver operating characteristic
library(pROC)
library(ggplot2)
roc1 <- roc(df$ pathologicaltype, df$consolidation)
roc2 <- roc(df$ pathologicaltype, df$waveletHLL_ngtdm_ Coarseness)
roc3 <- roc(df$ pathologicaltype, df$waveletLLH_glrlm_LongRunEmphasis)
roc4 <- roc(df$ pathologicaltype, df$waveletLHH_firstorder_Maximum)
roc5 <- roc(df$ pathologicaltype, df$PRE)
plot(roc1, legacy.axes = TRUE, add = T, col = 'blue', grid = c(0.5, 0.2))
plot(roc2, legacy.axes = TRUE, add = T, col = 'yellow')
plot(roc3, legacy.axes = TRUE, add = T, col = 'green')
plot(roc4, legacy.axes = TRUE, add = T, col = 'red')
plot(roc5, legacy.axes = TRUE, add = T, col = 'black')
legend = legend('bottomright', legend = c('consolidation
AUC=0.755', 'waveletHLL_ngtdm_Coarseness AUC=0.692',
'waveletHLL_firstorder_Maximum AUC=0.658',
'combined_training AUC=0.864'), col = c('blue', 'yellow',
'green', 'red', 'black'), cex = c(0.65), seg.len = 1, text.width = 0.85,
x.intersp = 0.1, y.intersp = 0.8, xpd = 'TURE', lwd = 2)

For nomogram
library('rms')
mod = lrm(t, training, x = T, y = T)
nom = nomogram(mod, fun = plogis, fun.at = c(0.1, seq(0.1, 0.9,
by = 0.3), 0.95),
lp = F, funlabel = 'Invasiveness Possibility')
plot(nom)

For computing the C-Index and 95% confidence interval (CI)
Library('Hmisc')
Cindex = rcorccens(training$pathologicaltype ~ predict(mod),
data = training)
95% CI, 1.96 x se; se = S.D./2

For calibration curve
Call1 = calibrate(mod, emethod = 'KM', method = 'boot', B = 1000)
par(mar = c(8, 8, 3, 2), cex = 0.8)
plot Call1, lwd = 2, lty = 1,
errbar.color = col = c(rgb(0, 0, 255, maxColorValue = 255)),
xlim = c(0, 1), ylim = c(0, 1),
xlab = 'Nomogram-Predicted Probability of Invasiveness',
ylab = 'Actual Invasiveness probability',
col = c(rgb(255, 0, 0, maxColorValue = 255)))

For decision curve analysis
library('rmda')
library('rms')
set.seed(123)
plot_decision_curve(list(clinical.model, radiomic.model,
combined.model),
curve.names = c('clinical_trainingl', 'radiomic_training',
'Combined_training'),
col = c('blue', 'red', 'green'), standardized = FALSE, lty = c(1, 1, 1),
lwd = c(3, 2, 2, 1), legend.position = 'bottomleft')
Figure S1. The distribution of the scores in the training set and validation set.