Applicability Domain (Williams Plot)

Saw Simeon, Nuttapat Anuwongcharoen, Watshara Shoombuatong, Aijaz Ahmad Malik, Virapong Prachayasittikul, Jarl E. S. Wikberg and Chanin Nantasenamat

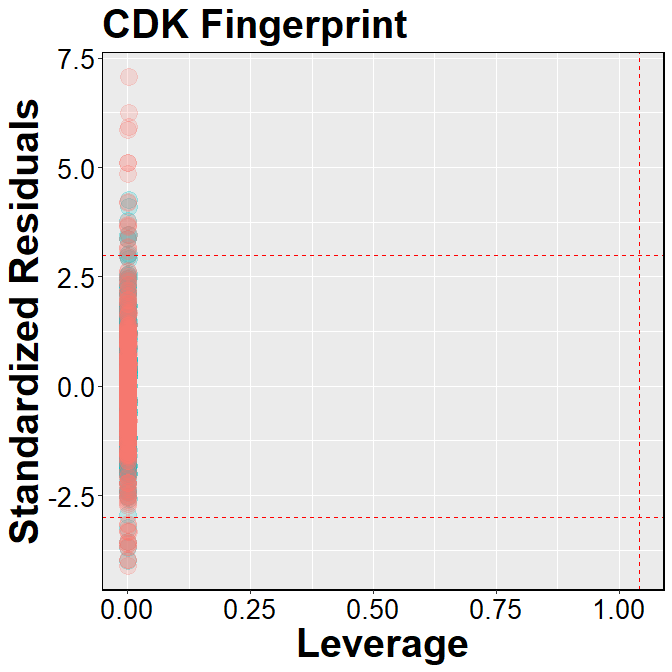
June 19, 2016

# Function to create Williams Plot (Red = External, Cyan = Internal)

file <- function(x) {  
 library(randomForest)  
 library(caret)  
 library(ranger)  
 set.seed(10)  
 para <- dplyr::sample\_n(x, size = 2186, replace = TRUE)  
 set.seed(3)  
 in\_train\_para <- sample(nrow(para),  
 size = as.integer(nrow(para) \* 0.8),  
 replace = FALSE)  
 set.seed(4)  
 Train <- para[in\_train\_para, ]  
 Test <- para[-in\_train\_para, ]  
   
 model <- ranger::ranger(pIC50~., data = Train, write.forest = TRUE, save.memory = TRUE)  
 #actual <- train$Activity  
 prediction <- predict(model, Train)  
 prediction\_Internal <- prediction$predictions  
 value <- data.frame(obs = Train$pIC50, pred = prediction\_Internal)  
 labeling <- c("obs", "pred")  
 colnames(value) <- labeling  
 value$Label <- c("Internal")  
 prediction\_External <- predict(model, Test)  
 prediction\_External <- prediction\_External$predictions  
 value\_external <- data.frame(obs = Test$pIC50, pred = prediction\_External)  
 colnames(value\_external) <- labeling  
 value\_external$Label <- c("External")  
 results <- rbind(value, value\_external)  
 return(results)  
}  
  
  
  
  
get\_leverage <- function(x) {  
 file <- file(x)  
 x <- file[, 1]  
 y <- file[, 2]  
 data <- data.frame(x, y)  
 error <- y-x  
 label <- file[3]  
 fit = lm(y~x,data = data)  
 hv <- as.data.frame(hatvalues(fit))  
 std.error = scale(error)  
 df <- data.frame(hv, std.error, label)  
 names(df) <- c("hv", "std.error", "Label")  
 return(df)  
}  
  
  
  
plot\_william <- function(x, title) {  
 library(ggplot2)  
 library(cowplot)  
 ok <- get\_leverage(x)  
 df <- data.frame(ok)  
 good <- ggplot(df, aes(hv, std.error)) +   
 geom\_point(aes(color = Label), alpha = .20, size = 6) +   
 ggtitle(title) +  
 xlab("Leverage") + ylab("Standardized Residuals") +   
 geom\_hline(yintercept = 3, color = "red", linetype = 2) +  
 geom\_hline(yintercept = -3, color = "red", linetype = 2) +  
 theme(  
 panel.border = element\_rect(linetype = "solid", colour = "black",  
 fill = NA, size = 1),  
  
 plot.title = element\_text(size = 30, color = "black", face = "bold"),  
 axis.text.y = element\_text(size = 20, colour = "black"),  
 axis.text.x = element\_text(size = 20, colour = "black"),  
 axis.title.x = element\_text(size = 30, color = "black", face = "bold"),  
 axis.title.y = element\_text(size = 30, color = "black", face = "bold"),  
   
 legend.position = ("none"))  
 return(good)  
   
}

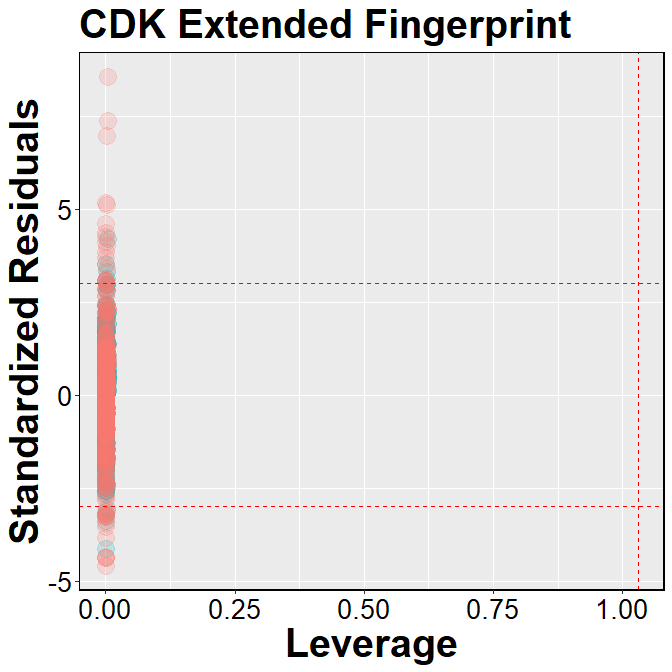
# CDK fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$FingerPrinter  
plot <- plot\_william(df, title = "CDK Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



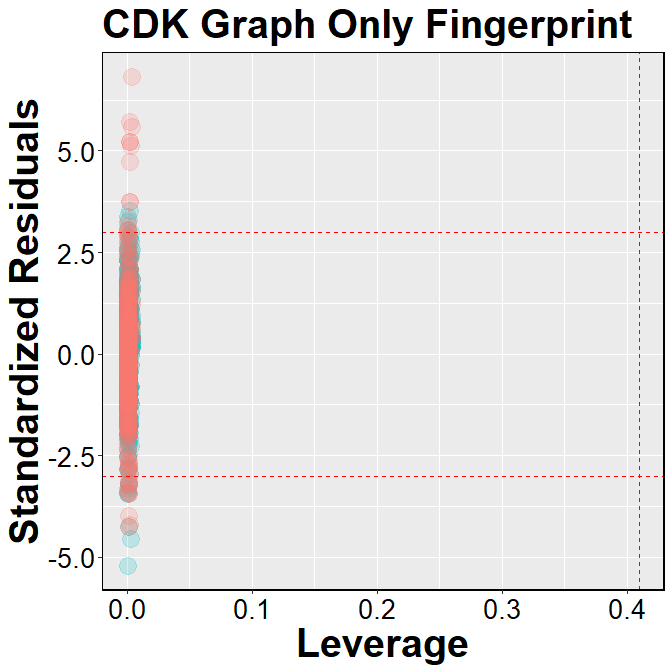
# CDK extended fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$Extended\_finterPrinter  
plot <- plot\_william(df, title = "CDK Extended Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



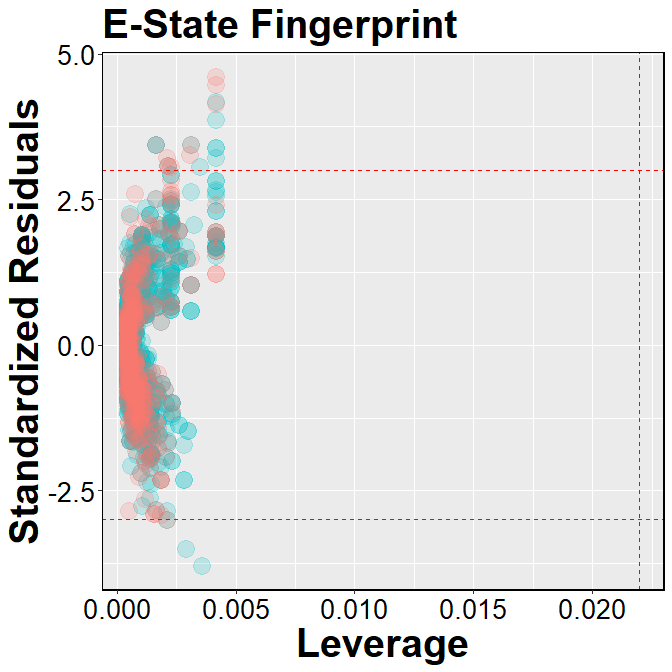
# CDK graph only fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$GraphOnly\_FingerPrinter  
plot <- plot\_william(df, title = "CDK Graph Only Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



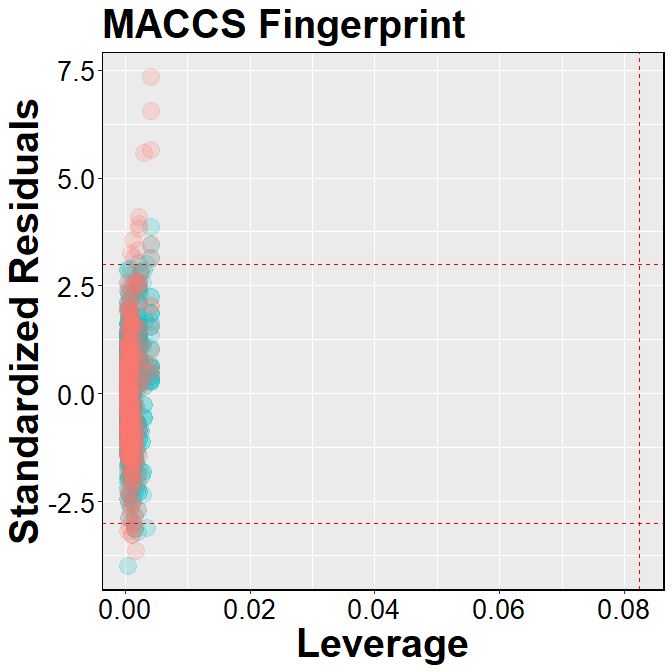
# E-State fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$Estate\_FingerPrinter  
plot <- plot\_william(df, title = "E-State Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



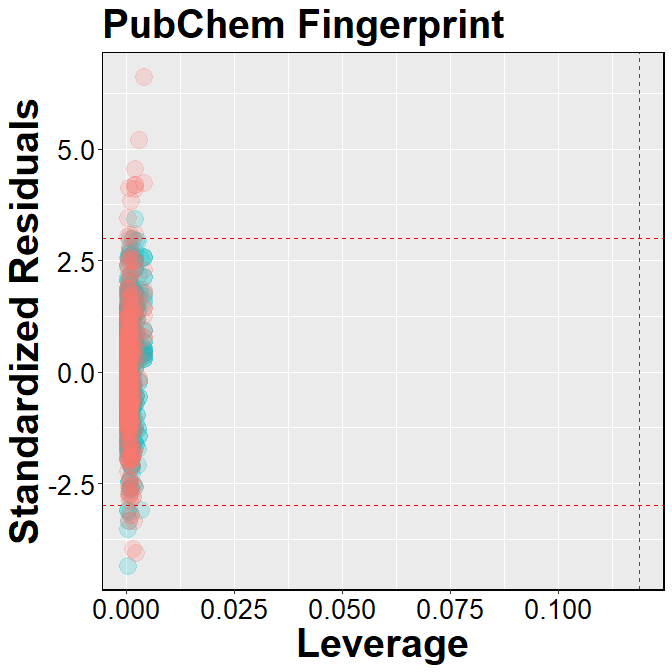
# MACCS fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$MACCS\_FingerPrinter  
plot <- plot\_william(df, title = "MACCS Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



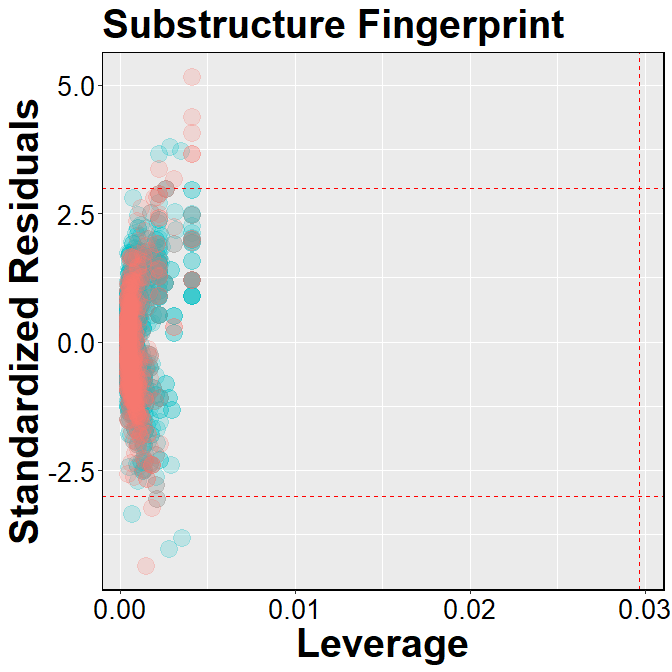
# PubChem fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$Pubchem\_FingerPrinter  
plot <- plot\_william(df, title = "PubChem Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



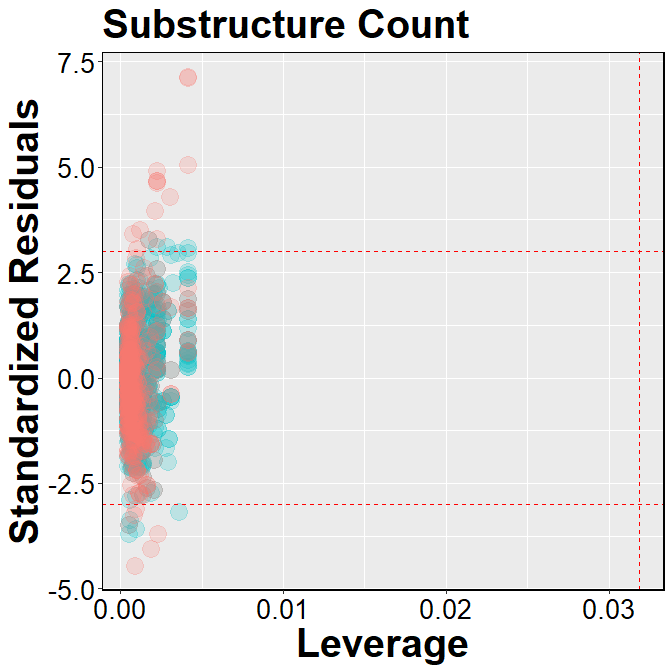
# Substructure fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$Substructure\_fingerPrinter  
plot <- plot\_william(df, title = "Substructure Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



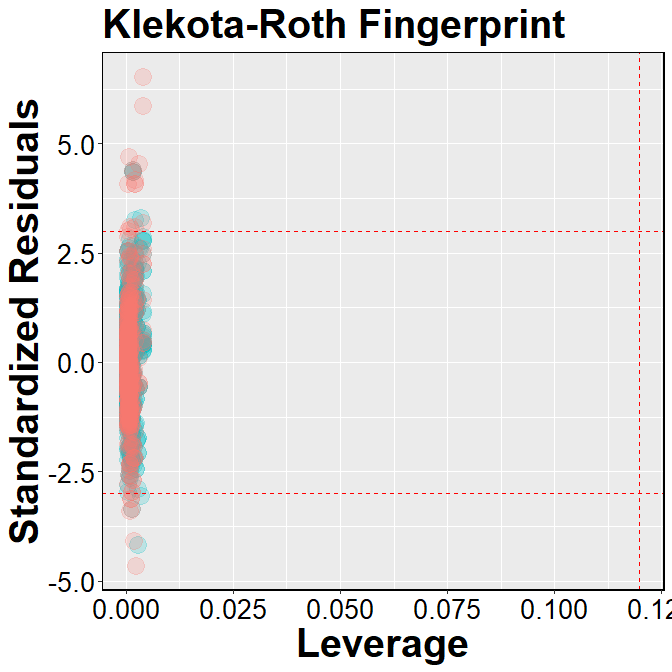
# Substructure count

input <- readRDS("fpdata.Rds")  
df <- input$Substructure\_fingerPrintCount  
plot <- plot\_william(df, title = "Substructure Count")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



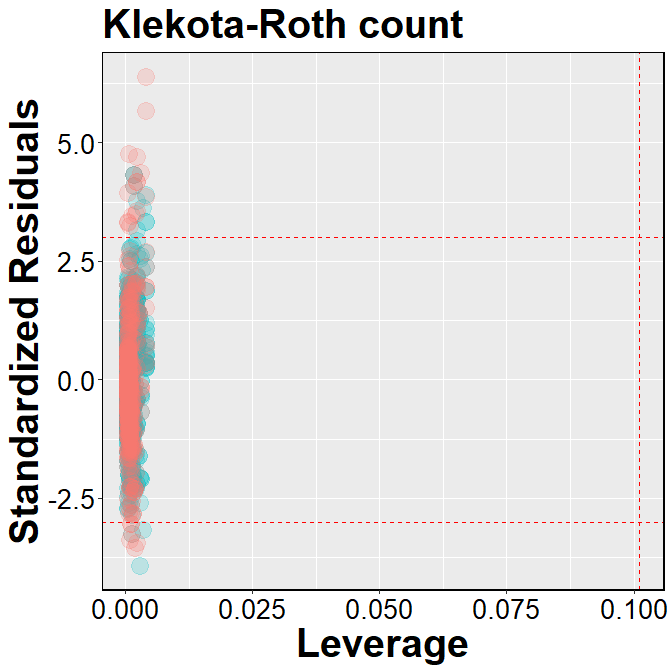
# Klekota-Roth fingerprint

input <- readRDS("fpdata.Rds")  
df <- input$KlekotaRoth\_FingerPrinter  
plot <- plot\_william(df, title = "Klekota-Roth Fingerprint")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



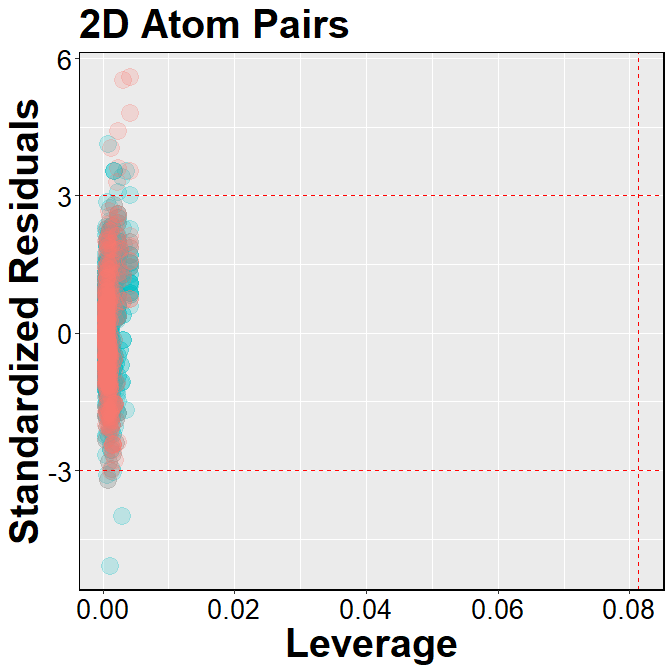
# Klekota-Roth count

input <- readRDS("fpdata.Rds")  
df <- input$KlekotaRoth\_FingerprintCount  
plot <- plot\_william(df, title = "Klekota-Roth count")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



# 2D atom pairs

input <- readRDS("fpdata.Rds")  
df <- input$AtomPairs2D\_fingerPrinter  
plot <- plot\_william(df, title = "2D Atom Pairs")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")



# 2D atom pairs count

input <- readRDS("fpdata.Rds")  
df <- input$AtomPairs2D\_fingerPrintCount  
plot <- plot\_william(df, title = "2D Atom Pairs Count")  
h <- 3\*((dim(df)[2] - 1) + 1) / dim(df)[1]\*0.8  
plot + geom\_vline(xintercept = h, linetype = 2, color = "red")

