Here is a brief explanation of the **PLAPLRML** package.

**Note :** Before using this R package, please ensure that the following four dependent R packages are installed, “glmnet”，”Matrix”，”gtools”，”ggplot2”.

**The main function:**

1. QPADMslack: the algorithm in Guan et al.2020 and Fan et al. 2021.

**The input and output of the main function:**

#' @param Pen 'LASSO' or 'SCAD' or 'MCP'.

#' @param lam Parameter tuning or regularization term parameters

#' @param X Matrix of predictors, of dimension (n\*p); each row is an observation

#' @param y Response variable

#' @param tau The quantile level τ and the value must be in (0,1)

#' @param M The number of local machines

#' @returns \item{beta\_u}{Regression coefficient.}

#' @returns \item{K}{number of iterations.}

#' @returns \item{t}{calculation time.}

**Regression and classification examples:**

#' ####### regression model

#' n <- 30000

#' p <- 1000

#' rho <- .5

#' beta\_true = rep(0, p)

#' beta\_true[6] = beta\_true[12] = beta\_true[15] = beta\_true[20] = 1

#' R <- matrix(0,p,p)

#' for(i in 1:p){

#' for(j in 1:p){

#' R[i,j] <- rho^abs(i-j)

#' }

#'}

#' X <- matrix(rnorm(n\*p),n,p) %\*% t(chol(R))

#' X[,1] = pnorm(X[,1])

#' tau=0.7

#' e = rnorm(n)

#' y = X[,6]+X[,12]+X[,15]+X[,20]+0.7\*X[,1]\*e

#' beta\_true[1] = 0.7\*qnorm(tau)

#' ##QPADMslack

#' modelslack=QPADMslack(X,y,Pen = "SCAD",lam=400\*sqrt(log(p)/n),tau=0.7,M=1) #

#' modelslack$beta\_u[c(1,6,12,15,20)]

#' modelslack$K

#' modelslack$t

#' length(which(abs(modelslack$beta\_u)>10^-4))

#' AE = sum(abs(modelslack$beta\_u - beta\_true))

#' AE

#' ##QPADMslackGB

#' modelslack=QPADMslackGB(X,y,Pen = "SCAD",lam=1200\*sqrt(log(p)/n),tau=0.7,M=1) #

#' modelslack$beta\_u[c(1,6,12,15,20)]

#' modelslack$K

#' modelslack$t

#' length(which(abs(modelslack$beta\_u)>10^-4))

#' AE = sum(abs(modelslack$beta\_u - beta\_true))

#' AE

#' ##QPADMslackGB2

#' modelslack=QPADMslackGB2(X,y,Pen = "SCAD",lam=1200\*sqrt(log(p)/n),tau=0.7,M=1) #

#' modelslack$beta\_u[c(1,6,12,15,20)]

#' modelslack$K

#' modelslack$t

#' length(which(abs(modelslack$beta\_u)>10^-4))

#' AE = sum(abs(modelslack$beta\_u - beta\_true))

#' AE

#' ##MQPADMslackGB

#' modelslack=MQPADMslackGB(X,y,Pen = "SCAD",lam=500\*sqrt(log(p)/n),tau=0.7,M=1) #

#' modelslack$beta\_u[c(1,6,12,15,20)]

#' modelslack$K

#' modelslack$t

#' length(which(abs(modelslack$beta\_u)>10^-4))

#' AE = sum(abs(modelslack$beta\_u - beta\_true))

#' AE

#' ##MQPADMslackGB2

#' modelslack=MQPADMslackGB2(X,y,Pen = "SCAD",lam=500\*sqrt(log(p)/n),tau=0.7,M=1) #

#' modelslack$beta\_u[c(1,6,12,15,20)]

#' modelslack$K

#' modelslack$t

#' length(which(abs(modelslack$beta\_u)>10^-4))

#' AE = sum(abs(modelslack$beta\_u - beta\_true))

#' AE

#' ####### classification model

#' n=5000

#' p=2000

#' q=10

#' rho <- 0.5 #Can be adjusted to 0.2, 0.4, 0.6

#' #First class

#' x1 <- matrix(rnorm(n\*q,0,1), n, q)

#' x2 <- matrix(rnorm(n\*(p-q),0,1), n, p-q)

#' corrmat1 <- toeplitz(rho^(0:(q-1)))

#' corrmat2 <- toeplitz(rho^(0:(p-q-1)))

#' X\_1 <- cbind(x1%\*% chol(corrmat1)+1, x2%\*% chol(corrmat2))

#' #Second class

#' x1 <- matrix(rnorm(n\*q,0,1), n, q)

#' x2 <- matrix(rnorm(n\*(p-q),0,1), n, p-q)

#' corrmat1 <- toeplitz(rho^(0:(q-1)))

#' corrmat2 <- toeplitz(rho^(0:(p-q-1)))

#' X\_2 <- cbind(x1%\*% chol(corrmat1)-1, x2%\*% chol(corrmat2))

#' ##Data preparation

#' X0= rbind(X\_1,X\_2)

#' X0 = cbind(matrix(1,nrow=2\*n,1),X0)

#' y\_label=c(rep(1,n),rep(-1,n))

#' ##Test dataset

#' nt=500

#' #First class

#' x1 <- matrix(rnorm(nt\*q,0,1), nt, q)

#' x2 <- matrix(rnorm(nt\*(p-q),0,1), nt, p-q)

#' corrmat1 <- toeplitz(rho^(0:(q-1)))

#' corrmat2 <- toeplitz(rho^(0:(p-q-1)))

#' X\_1 <- cbind(x1%\*% chol(corrmat1)+1, x2%\*% chol(corrmat2))

#' #Second class

#' x1 <- matrix(rnorm(nt\*q,0,1), nt, q)

#' x2 <- matrix(rnorm(nt\*(p-q),0,1), nt, p-q)

#' corrmat1 <- toeplitz(rho^(0:(q-1)))

#' corrmat2 <- toeplitz(rho^(0:(p-q-1)))

#' X\_2 <- cbind(x1%\*% chol(corrmat1)-1, x2%\*% chol(corrmat2))

#' X0t = rbind(X\_1,X\_2)

#' X0t = cbind(matrix(1,nrow=2\*nt,1),X0t)

#' yt\_label=c(rep(1,nt),rep(-1,nt))

#' n = nrow(X0)

#' p = ncol(X0)

#' y = rep(1,n)

#' X = diag(y\_label)%\*%X0

#' ##QPADMslack

#' SVMmodel=QPADMslack(X,y,Pen = "SCAD",lam=100\*sqrt(log(p)/n),tau=0.7,M=1)

#' (SVMmodel$beta\_u[1:11])/(SVMmodel$beta\_u[2])

#' SVMmodel$K

#' SVMmodel$t

#' length(which(abs(SVMmodel$beta\_u)>10^-4))

#' #Training set prediction accuracy

#' length(which((sign(X0%\*%SVMmodel$beta\_u)) - y\_label==0))/n

#' # Test set prediction accuracy

#' length(which((sign(X0t%\*%SVMmodel$beta\_u)) - yt\_label==0))/(2\*nt)

#' ##QPADMslackGB

#' SVMmodel=QPADMslackGB(X,y,Pen = "SCAD",lam=200\*sqrt(log(p)/n),tau=0.7,M=1)

#' (SVMmodel$beta\_u[1:11])/(SVMmodel$beta\_u[2])

#' SVMmodel$K

#' SVMmodel$t

#' length(which(abs(SVMmodel$beta\_u)>10^-4))

#' #Training set prediction accuracy

#' length(which((sign(X0%\*%SVMmodel$beta\_u)) - y\_label==0))/n

#' # Test set prediction accuracy

#' length(which((sign(X0t%\*%SVMmodel$beta\_u)) - yt\_label==0))/(2\*nt)

#' ##QPADMslackGB2

#' SVMmodel=QPADMslackGB2(X,y,Pen = "SCAD",lam=300\*sqrt(log(p)/n),tau=0.7,M=1)

#' (SVMmodel$beta\_u[1:11])/(SVMmodel$beta\_u[2])

#' SVMmodel$K

#' SVMmodel$t

#' length(which(abs(SVMmodel$beta\_u)>10^-4))

#' #Training set prediction accuracy

#' length(which((sign(X0%\*%SVMmodel$beta\_u)) - y\_label==0))/n

#' # Test set prediction accuracy

#' length(which((sign(X0t%\*%SVMmodel$beta\_u)) - yt\_label==0))/(2\*nt)

#' ##MQPADMslackGB

#' SVMmodel=MQPADMslackGB(X,y,Pen = "SCAD",lam=300\*sqrt(log(p)/n),tau=0.7,M=1)

#' (SVMmodel$beta\_u[1:11])/(SVMmodel$beta\_u[2])

#' SVMmodel$K

#' SVMmodel$t

#' length(which(abs(SVMmodel$beta\_u)>10^-4))

#' #Training set prediction accuracy

#' length(which((sign(X0%\*%SVMmodel$beta\_u)) - y\_label==0))/n

#' # Test set prediction accuracy

#' length(which((sign(X0t%\*%SVMmodel$beta\_u)) - yt\_label==0))/(2\*nt)

#' ##MQPADMslackGB2

#' SVMmodel=MQPADMslackGB2(X,y,Pen = "SCAD",lam=300\*sqrt(log(p)/n),tau=0.7,M=1)

#' (SVMmodel$beta\_u[1:11])/(SVMmodel$beta\_u[2])

#' SVMmodel$K

#' SVMmodel$t

#' length(which(abs(SVMmodel$beta\_u)>10^-4))

#' #Training set prediction accuracy

#' length(which((sign(X0%\*%SVMmodel$beta\_u)) - y\_label==0))/n

#' # Test set prediction accuracy

#' length(which((sign(X0t%\*%SVMmodel$beta\_u)) - yt\_label==0))/(2\*nt)