**Supplementary Methods**

***Liver function tests***

Concentrations of ALT, AST, alkaline phosphatase (ALP), total protein (TP), Alb, globulin, TBIL, DBIL, IBIL and TBAs in the rat serum were determined with an Automatic Biochemistry Analyzer (AU5400; Olympus, Tokyo, Japan).

***Analysis by R***

In order to analyze the differences in alterations of the *VEGF family members and their receptors* between tissues and species, and the potential roles in diagnosis and evaluations of HPS, we furtherly analyzed the expression matrix by R packages. All the statistics were conducted on R studio platform (version 1.4.1717).

***Heatmap for ELISA results by R***

To visualize the expression of VEGF family members and their receptors in specific tissues and species, the pheatmap package (version 1.0.12, https://cran.r-project.org/web/packages/pheatmap/index.html) 1 was applied to draw heatmaps. The fold change of expression of VEGF family members and their receptors in CEE negative, IPVD and HPS patients was calculated referring to healthy controls. Meanwhile the fold change of expression of VEGF family members and their receptors in experimental rats was calculated referring to the sham group. In this step, no cluster calculations were applied.

***Gardner-Altman Estimation plot for serum level of VEGF family members and their receptors in patients and experimental rats by R***

We intended to compare the serum level of VEGF family members and their receptors in patients with chronic liver disease and common bile duct ligation (CBDL) rat with healthy controls and sham rats, and visualize the mean differences, individual values and effect size. So that Gardner-Altman Estimation plot was applied. The tidyverse (version 1.3.1, https://cran.rstudio.com/web/packages/tidyverse/index.html) and dabestr 2 (version 0.3.0, https://cran.rstudio.com/web/packages/dabestr/index.html) package were used.

***Feature selection by Boruta and random forest (RF)***

RF is a typical machine learning algorithm for classification problems, which widely used in medical area. Boruta is a modified algorithm based on RF, which reduced the misleading impact of random fluctuation and correlation. To analyze the importance of the VEGF family members and their receptors in discriminating different stages of HPS development, Boruta 3 (version 7.0.0, https://cran.r-project.org/web/packages/Boruta/index.html) and caret 4 (version 6.0-90, https://cran.r-project.org/web/packages/caret/) were applied to do feature selection. After normalization, human data from center 1 were randomly divided into training (70%) and testing dataset (30%) by group (HPS and non-HPS) or group (CEE negative, IPVD and HPS), meanwhile data from center 2 were for externally validation. Feature selection was conducted on the training dataset. As for the results of importance ranking by RF, the importance was scaled so that the importance value of the first important factor was 100. For feature selection by Boruta, when the median of variable importance in the set runs was significantly higher or lower than the median of the maximum values for the shadow attribute (blue), the variable was confirmed important (green) or rejected as unimportant (red), otherwise tentatively important (yellow). Code was shown in the others section at the end of supplementary materials.

***Model construction and evaluation by RF based on four different inputs***

To investigate the role of clinical and ELISA results as variable to distinguish HPS from patients with chronic liver disease, we constructed the model on training dataset then evaluated on the testing dataset (the datasets were the same with those in the process of importance ranking). After normalization and one-hot coding of the raw data, different input factors were used for model construction (shown in Supplementary Table.2). When training, 10-fold cross validation is completely repeated 3 times. Afterwards, the model fitting was completed by RF. Confusion matrix (caret package) and ROC (pROC,5 version 1.18.0, https://cran.r-project.org/web/packages/pROC/index.html) were used for model evaluation. To investigate the overall improvement of the model between the Model 3 and the other ones, integrated discrimination improvement (IDI) was calculated and compared by PredictABEL package.6 Meanwhile, sensitivity, specificity, balanced accuracy (average of true positive rate and true negative rate), brier score, ROC (for two-classification) and multiclass ROC (for three-classification) were used.

**References**

1. Kolde R. pheatmap: Pretty Heatmaps. R package version 1.0.12. ed2019.

2. Joses Ho TT, Sameer Aryal, Hyungwon Choi, Adam Claridge-Chang. Moving beyond P values: Everyday data analysis with estimation plots. Nature Methods 2019;16: 1548-710. doi: 10.1038/s41592-019-0470-3.

3. Miron B. Kursa WRR. Feature Selection with the Boruta Package. Journal of Statistical Software 2010;36(11):1-13. doi: 10.18637/jss.v036.i11.

4. Kuhn M. caret: Classification and Regression Training. R package version 6.0-90. ed2021.

5. Xavier Robin NT, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez and Markus Müller. pROC: an open-source package for R and S+ to analyze and compare ROC curves . BMC Bioinformatics 2011;12:77. doi: 10.1186/1471-2105-12-77, PMID: 21414208.

6. Janssens SKaYSAaACJW. PredictABEL: Assessment of Risk Prediction Models. 2020. p. R package version 1.2-4.

**Supplementary materials**

**Antibodies**

|  |  |  |
| --- | --- | --- |
| **Name** | **Catalog number** | **Company** |
| anti-CD31 | ab119339 | Abcam |
| Cy3-conjuated Goat Anti-Mouse IgG (H+L) | 115165003 | Jackson |

**Main code for the current study**

***For data preprocessing***

library(dplyr)

library(caret)

# = = = = = normalized data = = = #

dat<-read.csv("data from xnyy.csv")

sdat = subset(dat,select = c("AaDO2","PaO2","SpO2","TBA","TBIL", "DBIL","ALB","ALT","AST","BUN","Cr","Hb","APTT","PT", "Fib","INR","PLT","Age","BMI","VEGF","VEGFR1","VEGFR2","MELD", "PLGF","sFlt1","sflt1\_PLGF"))

sdat<-lapply (sdat,scale)

sdat<-as.data.frame (sdat)

# = = = one-hot coding = = = #

Sex<-model.matrix(~Sex-1,dat) %>% as.data.frame()

hypertension<-model.matrix(~hypertension-1,dat) %>% as.data.frame()

diabetes<-model.matrix(~diabetes-1,dat) %>% as.data.frame()

trans<-cbind.data.frame(hypertension,diabetes,Sex)

post\_dat<-cbind.data.frame(trans,sdat)

post\_dat$class1 = dat[,2]

post\_dat$class2 = dat[,3]

post\_dat[,33]<-factor(post\_dat$class1,levels = c("0","1","2"),labels = c("neg","IPVD","HPS"))

post\_dat[,34]<-factor(post\_dat$class2,levels = c("0","1"),labels = c("nonHPS","HPS"))

# = = = = data splitting = = #

library(tidyverse)

library(caret)

set.seed(123)

training.samples1 <- post\_dat$class1 %>%

createDataPartition(p = 0.7, list = FALSE)

train.data1 <- post\_dat[training.samples1, ]

test.data1 <- post\_dat[-training.samples1, ]

set.seed (1234)

training.samples2 <- post\_dat$class2 %>%

createDataPartition(p = 0.7, list = FALSE)

train.data2 <- post\_dat[training.samples2, ]

test.data2 <- post\_dat[-training.samples2, ]

save(train.data1,train.data2,test.data1,test.data2,validation.data,file = "processed datasets.Rdata")

***For Bortua feature selection***

Library (Boruta)

load(file = "processed datasets.Rdata")

set.seed (12356)

boruta.train2 <-Boruta(class2~SpO2+MELD+TBA+TBIL+DBIL+ALB+ALT+AST+BUN+Cr+Hb+PT+APTT+Fib+INR+Age+BMI+hypertensionY+hypertensionN+diabetesY+diabetesN+SexF+SexM+VEGF+VEGFR1+VEGFR2+PLGF+sFlt1+sflt1\_PLGF, data = train.data2, pValue = 0.01,maxRuns = 150,doTrace = 2)

print(boruta.train2)

getSelectedAttributes (boruta.train2, withTentative = F)

train2\_df <- attStats (boruta.train2)

plot(boruta.train2, xlab = " ", xaxt = "n")

lz<-lapply(1:ncol(boruta.train2$ImpHistory),function(i)

boruta.train2$ImpHistory[is.finite(boruta.train2$ImpHistory[,i]),i])

names(lz) <- colnames(boruta.train2$ImpHistory)

Labels <- sort(sapply(lz,median))

axis(side = 1,las = 2,labels = names(Labels),

at = 1:ncol(boruta.train2$ImpHistory), cex.axis = 0.7)

final.boruta2 <- TentativeRoughFix(boruta.train2)

getConfirmedFormula(boruta.train2)

print(final.boruta2)

***For model fitting and evaluation***

load(file = "processed datasets.Rdata")

library(caret)

library(pROC)

# = = = = for crossvalidation on training datasets = = = #

control = trainControl(method = "repeatedcv",number = 10,repeats = 3)

traindata2<-cbind.data.frame(train.data2[,1:6],train.data2[,9:32])

set.seed(12345)

model2<-train(x = traindata2,y = train.data2$class2,method = "rf",trControl = control)

# = = = = = importance by RF = = = #

importance<-varImp(model2,scale = T)

plot(importance)

# = = = = = = = = = = = = = = = = = = = = below is for two-classification probelems = = = = = = = = ##

# = = = calculate the prediction value = = = #

pre2\_test<-predict(model2,test.data2)

pre2\_test2<-predict(model2,test.data2,type = "prob")

pre2\_vali<-predict(model2,validation.data)

pre2\_vali2<-predict(model2,validation.data,type = "prob")

# = = = = = confusion matrix = = = #

conmax2\_test<-confusionMatrix(pre2\_test,test.data2$class2,mode = "everything",positive = "HPS")

conmax2\_vali<-confusionMatrix(pre2\_vali,validation.data$class2,mode = "everything",positive = "HPS")

rocs2\_test<-roc(response = test.data2$class2,predictor = pre2\_test2[,2],plot = T,smooth = F)

auc(rocs2\_test)

ci(rocs2\_test)

rocs2\_vali<-roc(response = validation.data$class2,predictor = pre2\_vali2[,2],plot = T,smooth = F)

auc(rocs2\_vali)

ci(rocs2\_vali)

# = = = draw ROC curves processed with smooth = = = #

rocs\_vali<-roc(response = validation.data$class2,predictor = pre2\_vali2[,2],plot = T,smooth = T,,ci = T,col = "brown",print.auc = TRUE,legacy.axes = T,print.auc.x = 1.0,print.auc.y = 0.8,xlim = c(1,0),ylim = c(0,1))

rocs\_test<-roc(response = test.data2$class2,predictor = pre2\_test2[,2],plot = T,smooth = T,ci = T,col = "navy",print.auc = TRUE,legacy.axes = T,print.auc.x = 1.0,print.auc.y = 0.6,xlim = c(1,0),ylim = c(0,1),add = T)

legend(1,0.6,bty = "n",legend = c("Test","Validation"),col = c("navy","brown"),cex = 0.8,lwd = 2)

# = = = = = = = = = = = = = = = = = = = = = = = below is for three-classification problems = = = = = = = ##

pre1\_test<-predict(model1,test.data1)

pre1\_test1<-predict(model1,test.data1,type = "prob")

conmax1\_test<-confusionMatrix(pre1\_test,test.data1$class1,mode = "everything")

pre1\_vali<-predict(model1,validation.data)

pre1\_vali1<-predict(model1,validation.data,type = "prob")

conmax1\_vali<-confusionMatrix(pre1\_vali,validation.data$class1,mode = "everything")

rocs1\_test<-multiclass.roc(response = test.data1$class1,predictor = pre1\_test1[,3],plot = T,smooth = F)

auc(rocs1\_test)

***For heatmap***

***# = = = = = = = = =*** = For Elisa ***= = = = = = = = = = = = = = #***

library(pheatmap)

oridat<-read.csv("elisa human.csv",header = T,stringsAsFactors = T)

sub\_dat<-subset (oridat,select = c("VEGF","VEGFR1","VEGFR2","plgf","sflt1","sflt\_plgf"))

nor\_sub\_dat<-sapply(sub\_dat,scale)

summary(nor\_sub\_dat)

processed\_dat<-nor\_sub\_dat

annotation\_row = data.frame(group = factor(rep(c("Healthy","CEE\_neg","IPVD","HPS"), c(6,33,31,41))))

annotation\_col = c("VEGF","VEGFR1","VEGFR2","PLGF","sFlt1","sFlt1\_PLGF")

annotation\_col <-as.data.frame(annotation\_col)

rownames(annotation\_row) = paste("sample", 1:111, sep = "")

rownames(annotation\_col) = c("VEGF", "VEGFR1", "VEGFR2","PLGF","sFlt-1","sFlt-1/PLGF")

rownames(processed\_dat) = rownames(annotation\_row)

colnames(processed\_dat) = rownames(annotation\_col)

ann\_colors = list(group = c(Healthy = "navy",CEE\_neg = "grey",IPVD = "orange",HPS = "red"))

n = processed\_dat[,1:6]

n[n>2] = 2

n[n< -2] = -2

n[1:4,1:4]

pheatmap(n,annotation\_row = annotation\_row,annotation\_colors = ann\_colors,cellwidth = 25,cellheight = 4,fontsize\_row = 10,fontsize\_col = 14,border\_color = NA,cluster\_cols = F,cluster\_rows = F,show\_rownames = F,show\_colnames = T)

# = = = = = = = = = = = = = = = = For fold change = = = = = = = = = = = = = = = = = = #

library(pheatmap)

dat<-read.csv("elisa human+rat compare to sham.csv",header = T,stringsAsFactors = T)

dat<-dat[,-1]

dat<-as.matrix(dat)

rownames(dat) = c('VEGF','VEGFR1','VEGFR2','PLGF','sFlt-1','sFlt-1/PLGF')

pheatmap(dat,show\_rownames = T,display\_numbers = T,cellwidth = 20,cellheight = 15,cluster\_rows = F,cluster\_cols = F)

annotation\_col = data.frame(group = factor(rep(c("human\_serum","rat\_serum","rat\_liver","rat\_lung"), c(3,3,3,3))))

annotation\_row = c('VEGF','VEGFR1','VEGFR2','PLGF','sFlt-1','sFlt-1/PLGF')

annotation\_row <-as.data.frame(annotation\_row)

rownames(annotation\_row) = c('VEGF','VEGFR1','VEGFR2','PLGF','sFlt-1','sFlt-1/PLGF')

rownames(annotation\_col) = c('cee\_neg','IPVD','HPS','s\_CBDL3w','s\_CBDL5w', 'li\_CBDL3w','li\_CBD5w', 'lu\_CBDL3w','lu\_CBDL5w')

rownames(dat) = rownames(annotation\_row)

colnames(dat) = rownames(annotation\_col)

ann\_colors = list(group = c(human\_serum = red",rat\_serum = "yellow",rat\_liver = "green",rat\_lung = "brown"))

pheatmap(dat,annotation\_col = annotation\_col,annotation\_colors = ann\_colors,border\_color = NA,fontsize\_number = 10,

show\_rownames = T,show\_colnames = T,display\_numbers = T,number\_color = "black",number\_format = "%.3f",cellwidth = 30,cellheight = 20,cluster\_rows = F,cluster\_cols = F)

pheatmap(dat,show\_rownames = T,display\_numbers = T,number\_color = "black",border\_color = NA,fontsize\_number = 14,number\_format = "%.3f",cellwidth = 50,cellheight = 40,cluster\_rows = F,cluster\_cols = F)

# = = = = = = = = = = = For Gardner-Altman Estimation plot = = = = = = = = = = = = = = = = #

library(tidyverse)

library(dabestr)

data<-read.csv("rat and human serum.csv",header = T, stringsAsFactors = T)

multi.group <-

data %>%

dabest(group, VEGFR2,

idx = list(c("h\_healthy","h\_CEEneg", "h\_IPVD", "h\_HPS"),

c("sham", "CBDL1w", "CBDL3w","CBDL5w")),

paired = FALSE)

multi.group.mean\_diff <- multi.group %>% mean\_diff()

multi.group.mean\_diff

plot(multi.group.mean\_diff, color.column = species,

rawplot.markersize = 3,rawplot.groupwidth = 0.4)