Phenopro Manual

Sajjad Fouladvand, Jin Chen

Chen Lab, Institute for Biomedical Informatics, University of Kentucky, Lexington, KY

Email: sfo247@g.uky.edu

The source code is available using: https://github.com/sajjadfouladvand/PhenoPro7

1. Package installation

Install PhenoPro version 7 using the following commands:

```
install.packages("devtools")

library(devtools)

install_github("sajjadfouladvand/PhenoPro7")

library(PhenoPro7)
```

Although required packages should be automatically added while you installed PhenoPro, we recommend you run the following lines of code to make sure you have all required packages loaded:

```
library(ggplot2)

library(MASS)

library(e1071)

library(randomForest)

library(gridExtra)

library(gtools)

library(viridis)

library(ggthemes)

library(FSelector)

library(xgboost)
```

2. Load sample data and test package installation

This manual use RN data set. Following lines of code extract RN Early Resistant data out of entire RN data set, however, you can select each parts of the data you want to analysis:

```
setwd("directory of your data")

data_all <- read.table("RN.csv", header = T, sep=",")
```

```
data_clean <- data_all[which(data_all$Issues == 0),]

data <- subset (data_clean, (Stage == "WPI1"| Stage == "WPI2") & (Entry==885))

data_new <- PhenoPro7::BlockSplit(x = data, blockName = c("BlkTempC","PlantC"),

                    label = "Treatment", discard = FALSE)
```

Then, you can randomly divide the data set into train and test set using:

```
data_all_sep <- PhenoPro7::train.test.generation(data=data_new,x=c("AmbientHumidity",
"LeafTemp", "LeafAngle", "LEF", "LightIntensityPAR"), y=c("NPQt",
"Phi2","PhiNO","PhiNPQ","RelativeChlorophyll"), label="Treatment", defaultLabel = "D",
block= c("NewBlockNameOne","NewBlockNameTwo"), orderby ="TimeofDay", testBlockProp
= 0.1 )
```

After generating training and test sets, you need to train PhenoPro using the training data:

```
res <- PhenoPro7::Pheno(data = data_all_sep$train.set,

         x     =c("AmbientHumidity",   "LeafTemp",    "LeafAngle",    "LEF",
"LightIntensityPAR","NPQt", "Phi2","PhiNO","PhiNPQ","RelativeChlorophyll"),

         y    =    c("AmbientHumidity",   "LeafTemp",    "LeafAngle",    "LEF",
"LightIntensityPAR","NPQt", "Phi2","PhiNO","PhiNPQ","RelativeChlorophyll"),

              label = "Treatment", defaultLabel = "D",

              block = c("NewBlockNameOne", "NewBlockNameTwo"),

              orderby = "TimeofDay",

              method = "SVM", step = 1, width = 6, nfeatures = 10,    feature_selection  =
              "IG")
```

The function "*Pheno*" has several arguments as its inputs:

- data: Is the input data that will be used to train the model. The default value is NULL.
- x: Represent the variables to be paired with variables in *y(see next parameter).* The default value is NULL.
- y: Represent the variables to be paired with variables in *x.* The default value is NULL.
- label: Indicates the variable that we are aiming to predict using PhenoPro. The default value is NULL.
- block: Is a variable that shows block IDs. The default value is NULL.
- orderBy: Shows the variable which should be used t order the samples. Normally, it should a time based variable.
- method: Determines which classification method should be used to create the predictive model. It can be either "SVM" for using support vector machines classifier or "RF" for Random Forest classifier or "Ensemble" for combination of SVMs. The default value is "SVM".

- step: The default value is 1.
- width: The size of the window will be 2 multiples by *width*. The default value is 6.
- nfeatures: It defines either the number of principal components when "PCA" is used as dimensionality reduction method or the number of top features that PhenoPro should use when a feature selection method is used (refer to the next argument, feature_selection). The default value is 3.
- feature_selection: It defines the feature selection method which will be used to find the optimal subset of features. The options are: "lmFuncs" for linear regression based feature subset selection, "rfFuncs" for random forest based feature subset selection, "treeBagFuncs" for bagged tree based feature subset selection, "bnFuncs" for naïve bayes based feature subset selection, "PCA" for principle component analysis, "IG" for information gain based method, "Ensemble of SVMs" for a pair-wise feature selection method using SVM, and "NA" for no feature selection which is in fact using the whole feature set. The default value is "lmFuncs".

The trained model should be tested using unseen samples, therefore we need to separate the labels from testing data set:

unseen_data<- data_all_sep$test.set[, !names(data_all_sep$test.set) %in% c("Treatment")]

We then send this unseen testing set to PhenoPro to predict the labels:

cvres<- PhenoPro7::test.unseen.pheno(WorkingData =unseen_data , predictive_model = res$pre,
pca_model = res$resPCA, scales_matrix = res$scales_matrix,
weak_classifiers=res$weak_classifiers, bst = res$bst,label_mapping = res$label_mapping ,
nfeatures=res$nfeatures,feature_selection = res$feature_selection,block = resargblock,
block_orig= c("NewBlockNameOne", "NewBlockNameTwo"),orderby = resargorderby,
step=1, width=6, method = resargmethod, defaultLabel = resargdefaultLabel,
topFeatureFullNames = res$feaf, topFeatureNames = res$fea, x=res$arg$x, y=res$arg$y, prior = FALSE)

Eventually we give the PheoPro package the predicted labels and actual labels to measure the performance on the testing data.

perf<- PhenoPro7::calc.performance(cvres$blocks_labels_preds,
data_all_sep$test.set$Treatment, pos_label="D")

Now, the object *perf* include various measurements:

- perf$sample_base_res: Accuracy, Precision and Recall calculated considering all samples **within** testing blocks.
- perf$block_based_res_detail: Accuracy, Precision and Recall calculated considering each testing block as one sample (a voting schema is used to label testing blocks).

Please let me know if you have any questions about how PhenoPro7 package.