Machine learning identifies pathophysiologically and prognostically informative phenotypes among patients with mitral regurgitation undergoing transcatheter edge-to-edge repair

Code explanation for future patient–to-cluster assignment

```r
##########################################################################################################################
# Loading basic libraries (standard):
library(tidyverse)
library(gcookbook)
library(readxl)
library(ggpubr)
library(ggExtra)
library(dplyr)
library(ggbeeswarm)


##########################################################################################################################
# At first, please import the data on which the clustering approach is based on:

R_Code_backbone_data <- read_excel("Desktop/MitraClip Trenkwalder 2022 Revision EHJ_CVI 2 Future Assignment/R_Code_backbone.xlsx")

head(R_Code_backbone_data)
dim(R_Code_backbone_data)
sum(is.na(R_Code_backbone_data))
colSums(is.na(R_Code_backbone_data))


# Next, please characterize the future patient according to the requested input parameters for the artificial neural network:
new_patient <- data.frame(
  LVEF_pre = 51.5,                          # left ventricular ejection fraction in %
  LVESD_pre = 41.0,                         # left ventricular end-systolic diameter in mm
  EROA_pre = 0.438,                         # mitral valve effective regurgitant orifice area in cm2
  sPAP_pre = 48.9,                          # systolic pulmonary artery pressure (as assessed by echocardiography) in mmHg
  TAPSE_pre = 17.1,                         # tricuspid annular plane systolic excursion in mm
  RV_Mitte_pre = 33.6,                      # right midventricular diameter in mm
  LA_volume_pre = 312,                      # left atrial volume in ml
  RA_size_pre = 46.0                        # right atrial area in cm2
)


# In case that some input parameters from the future patient should be missing (e.g. sPAP levels), those can be imputed:
library(missForest)
set.seed(104)

data1_selected <- rbind(new_patient, R_Code_backbone_data)  # In order to impute missing values, the derivation cohort can be used as additional information.
dim(data1_selected)

data1_selected <- as.data.frame(data1_selected)   # Convert to a dataframe for missForest
data1_selected.imp <- missForest(data1_selected)  # From now on, missing values are imputed
data1_selected.imp <- data1_selected.imp$ximp     # To save only the data matrix with imputed values as a new data frame (without estimated imputation errors)


# Scaling of selected variables:
data1_selected.imp.scaled <- scale(data1_selected.imp)
data1_selected.imp.scaled <- as.data.frame(data1_selected.imp.scaled)

data1_selected.imp.scaled <- data1_selected.imp.scaled[1:1, 1:8]   # Only keep the future patient for patient-to-cluster assignment
head(data1_selected.imp.scaled)
dim(data1_selected.imp.scaled)

##########################################################################################################################
# Load necessary libraries for future patient-to-cluster assignment:
library(caret)
library(keras)
library(tensorflow)

# Load the trained artificial neural network:
Trained_ANN <- load_model_hdf5("Desktop/MitraClip Trenkwalder 2022 Revision EHJ_CVI 2 Future Assignment/MitraClip_ANN_prospective_assignment.h5")

# Now, please assign the future patient to a cluster:
Cluster_assignment <- Trained_ANN %>% predict_classes(as.matrix(data1_selected.imp.scaled))

Cluster_assignment <- case_when(
  Cluster_assignment == 0  ~ "Cluster 1",
  Cluster_assignment == 1  ~ "Cluster 2",
  Cluster_assignment == 2  ~ "Cluster 3",
  Cluster_assignment == 3  ~ "Cluster 4"
)

print(Cluster_assignment)

##########################################################################################################################
```

functional and structural parameters from a future patient serving as input data to the trained artificial neural network

```
> # Now, please assign the future patient to a cluster:
> Cluster_assignment <- Trained_ANN %>% predict_classes(as.matrix(data1_selected.imp.scaled))
>
> Cluster_assignment <- case_when(
+   Cluster_assignment == 0  ~ "Cluster 1",
+   Cluster_assignment == 1  ~ "Cluster 2",
+   Cluster_assignment == 2  ~ "Cluster 3",
+   Cluster_assignment == 3  ~ "Cluster 4"
+ )
>
> print(Cluster_assignment)
[1] "Cluster 4"
>
> ##########################################################################################################################
>
```

model output, i.e. cluster assignment for the future patient