SUPPLEMENTARY FILE 2

Random Forest further modelling

1. Random Forest – best *ntree*

```
#load the package
     >library (randomForest)

#read file
     >all_data <- read.csv (file='D:/model_evaluation/all_data.csv')

#run randomForest
     >all_data.rf <- randomForest (V24 ~. , data=all_data, ntree=500,
     mtry=5, keep.forest=FALSE, importance=TRUE)

#print result of randomForest (OOB error and confusion matrix)
     >print (all_data.rf)

#find best ntree for Random Forest
     >which.min(all_data.rf$err.rate[,1])
```

---

2. Random Forest Further Modelling – for all clusters

Repeat the steps in (1) with different clusters of data to determine the accuracy of the model

3. Random Forest – calibration plot using *Phyton 3*

```
#Import packages
   import numpy as np
   import pandas as pd
   import matplotlib.pyplot as plt
   from sklearn import preprocessing
   from sklearn import model_selection

#Import dataset
   x_train_file = 'x_train.xlsx'
   y_train_file = 'y_train.xlsx'
   def my_data(file_name):
       data = np.array(pd.read_excel(file_name, 'Sheet1'))
       scaler=preprocessing.MinMaxScaler()
       data=scaler.fit_transform(data)
       data=data.astype('float32')
       return data

#split the data into training and test set
   bc_X_train, bc_X_test, bc_y_train, bc_y_test = train_test_split(
   my_data(x_train_file),my_data(y_train_file),test_size=0.2)

#Import RandomForestClassifier package
      from sklearn.ensemble import RandomForestClassifier
      rf_model = RandomForestClassifier(random_state=1234).fit(X= bc_X_train,
      y= bc_y_train)
      rf_prediction = rf_model.predict_proba(bc_X_test)

#Import calibration curve package
      from sklearn.calibration import calibration_curve

#compute calibration curve
      rf_y, rf_x = calibration_curve(bc_y_test, rf_prediction[:,1],
      n_bins=10)

#Plot the calibration lines
      import matplotlib.pyplot as plt
      import matplotlib.lines as mlines
      import matplotlib.transforms as mtransforms
      fig, ax = plt.subplots()

#only this line is calibration curve
      plt.plot(rf_x, rf_y, marker='o', linewidth=1, label='rf')

#reference line, legends, and axis labels
      line = mlines.Line2D([0, 1], [0, 1], color='black')
      transform = ax.transAxes
      line.set_transform(transform)
      ax.add_line(line)
      fig.suptitle('Calibration plot for Breast Cancer data')
      ax.set_xlabel('Predicted survival')
      ax.set_ylabel('True probability in each bin')
      plt.legend()
      plt.show()
```