In [1]:
```
#The following code was tested using python 3.7.6


import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import (brier_score_loss, precision_score, recall_score,
                             f1_score)
from sklearn.model_selection import StratifiedShuffleSplit
```

In [12]:
```
#Secting the chain lengths from modern plants
X = #array of relative abundance of n chain lengths from modern plants normalized to unity

#Selecting the labels of the modern plants
y = #array of modern plant classes of each sample in X

#Selecting the core wax chain length
z= #array of relative abundance of n chain lengths from core waxes normalized to unity
```

In [16]:
```
#lists for accuracy scores of ML algorithms

RF_scores = []
gpc_scores = []
log_scores = []
SVM_scores = []
net_scores = []

#lists of F scores of ML algorithms
RF_fscore = []
SVM_fscore = []
gpc_fscore = []
log_fscore = []
net_fscore = []

#lists of macrophyte vegetation probability over all train/test splits
gpc_suffle_core_m=[]
log_suffle_core_m=[]
RF_suffle_core_m=[]
net_suffle_core_m=[]
SVM_suffle_core_m=[]

#lists of conifer vegetation probability over all train/test splits
gpc_suffle_core_c=[]
log_suffle_core_c=[]
RF_suffle_core_c=[]
net_suffle_core_c=[]
SVM_suffle_core_c=[]

#lists of desert vegetation probability over all train/test splits
gpc_suffle_core_d=[]
log_suffle_core_d=[]
RF_suffle_core_d=[]
net_suffle_core_d=[]
SVM_suffle_core_d=[]

#Creating train and test sets from our data
sss=StratifiedShuffleSplit(n_splits=5, test_size=0.34)

#Looping the ML algorithms over each train and test split
for train_index, test_index in sss.split(X, y):

        X_train, X_test, y_train, y_test = X[train_index], X[test_index], y[train_index], y[test_index]

        #defining and fitting the random forest model to training data
        RF_model=RandomForestClassifier(n_estimators=300,n_jobs=-1).fit(X=X_train, y=y_train)

        #defining and fitting SVM to training data
        SVM_model=SVC(probability=True,C=9.59,gamma=2.7).fit(X=X_train, y=y_train)

        #defining and fitting the Gaussian Process Classifier to training data
        kernel = 1.0 * RBF(1.0)
        gpc_model = GaussianProcessClassifier(kernel=kernel).fit(X=X_train, y=y_train)

        #defining and fitting the logistic regression classifier to training data
        log_model=LogisticRegression(C=35,multi_class='multinomial').fit(X_train, y_train)

        #defining and fitting the neural network to training data
        net_model=MLPClassifier(max_iter=2000,hidden_layer_sizes=(100,100,100)).fit(X_train, y_train)

        #Calculating model accuracy
        RF_scores.append(RF_model.score(X_test, y_test))
        gpc_scores.append(gpc_model.score(X_test, y_test))
        SVM_scores.append(SVM_model.score(X_test, y_test))
        log_scores.append(log_model.score(X_test, y_test))
        net_scores.append(net_model.score(X_test, y_test))

        #Calculating model F1 score
        SVM_fscore.append(f1_score(y_test, SVM_model.predict(X_test), average='macro'))
        gpc_fscore.append(f1_score(y_test, gpc_model.predict(X_test), average='macro'))
        RF_fscore.append(f1_score(y_test, RF_model.predict(X_test), average='macro'))
        log_fscore.append(f1_score(y_test, log_model.predict(X_test), average='macro'))
        net_fscore.append(f1_score(y_test, net_model.predict(X_test), average='macro'))


        #Predicting conifer probability from core data
        SVM_suffle_core_c.append(SVM_model.predict_proba(z)[:,0])
        gpc_suffle_core_c.append(gpc_model.predict_proba(z)[:,0])
        RF_suffle_core_c.append(RF_model.predict_proba(z)[:,0])
        log_suffle_core_c.append(log_model.predict_proba(z)[:,0])
        net_suffle_core_c.append(net_model.predict_proba(z)[:,0])

        #Predicting desert plant probability from core data
        SVM_suffle_core_d.append(SVM_model.predict_proba(z)[:,1])
        gpc_suffle_core_d.append(gpc_model.predict_proba(z)[:,1])
        RF_suffle_core_d.append(RF_model.predict_proba(z)[:,1])
        log_suffle_core_d.append(log_model.predict_proba(z)[:,1])
        net_suffle_core_d.append(net_model.predict_proba(z)[:,1])

        #Predicting macrophyte plant probability from core data
        SVM_suffle_core_m.append(SVM_model.predict_proba(z)[:,2])
        gpc_suffle_core_m.append(gpc_model.predict_proba(z)[:,2])
        RF_suffle_core_m.append(RF_model.predict_proba(z)[:,2])
        log_suffle_core_m.append(log_model.predict_proba(z)[:,2])
        net_suffle_core_m.append(net_model.predict_proba(z)[:,2])




print('SVM accuracy -', np.mean(SVM_scores))
print('gpc accuracy -',np.mean(gpc_scores))
print('RF accuracy -',np.mean(RF_scores))
print('Log reg accuracy -',np.mean(log_scores))
print('Net accuracy -',np.mean(net_scores))

print('SVM F1 -',np.mean(SVM_fscore))
print('gpc F1 -',np.mean(gpc_fscore))
print('RF F1 -',np.mean(RF_fscore))
print('Log reg F1 -',np.mean(log_fscore))
print('Net F1 -',np.mean(net_fscore))

#Joining seperate train test splits of core predictions
RF_joined_conifer=np.vstack(RF_suffle_core_c)
Log_joined_conifer=np.vstack(log_suffle_core_c)
GPC_joined_conifer=np.vstack(gpc_suffle_core_c)
SVM_joined_conifer=np.vstack(SVM_suffle_core_c)
net_joined_conifer=np.vstack(net_suffle_core_c)

RF_joined_desert=np.vstack(RF_suffle_core_d)
Log_joined_desert=np.vstack(log_suffle_core_d)
GPC_joined_desert=np.vstack(gpc_suffle_core_d)
SVM_joined_desert=np.vstack(SVM_suffle_core_d)
net_joined_desert=np.vstack(net_suffle_core_d)

RF_joined_macrophyte=np.vstack(RF_suffle_core_m)
Log_joined_macrophyte=np.vstack(log_suffle_core_m)
GPC_joined_macrophyte=np.vstack(gpc_suffle_core_m)
SVM_joined_macrophyte=np.vstack(SVM_suffle_core_m)
net_joined_macrophyte=np.vstack(net_suffle_core_m)

#Calculating mean model core prediction from seperate train/test splits
RF_mean_conifer=RF_joined_conifer.mean(axis=0)
Log_mean_conifer=Log_joined_conifer.mean(axis=0)
GPC_mean_conifer=GPC_joined_conifer.mean(axis=0)
SVM_mean_conifer=SVM_joined_conifer.mean(axis=0)
net_mean_conifer=net_joined_conifer.mean(axis=0)

RF_mean_desert=RF_joined_desert.mean(axis=0)
Log_mean_desert=Log_joined_desert.mean(axis=0)
GPC_mean_desert=GPC_joined_desert.mean(axis=0)
SVM_mean_desert=SVM_joined_desert.mean(axis=0)
net_mean_desert=net_joined_desert.mean(axis=0)

RF_mean_macrophyte=RF_joined_macrophyte.mean(axis=0)
Log_mean_macrophyte=Log_joined_macrophyte.mean(axis=0)
GPC_mean_macrophyte=GPC_joined_macrophyte.mean(axis=0)
SVM_mean_macrophyte=SVM_joined_macrophyte.mean(axis=0)
net_mean_macrophyte=net_joined_macrophyte.mean(axis=0)
```

```
SVM accuracy - 0.8352941176470587
gpc accuracy - 0.8235294117647058
RF accuracy - 0.8235294117647058
Log reg accuracy - 0.8470588235294118
Net accuracy - 0.8235294117647058
SVM F1 - 0.8276819476819476
gpc F1 - 0.798011988011988
RF F1 - 0.8174825174825173
Log reg F1 - 0.8348910348910348
Net F1 - 0.8133688533688532
```