Microarray Analysis - Machine Learning

Author: Tony Kabilan Okeke

In this study, you will analyze a Breast Cancer dataset GSE7390, and identify a gene signature for prediction of Breast Cancer relapse.
Use SVM to predict relapse. Use a forward-selection strategy and 10-fold crossvalidation to determine the best gene signature.

In [ ]:
```
%load_ext autoreload
```

In [ ]:
```
# Imports
%autoreload 2
import pandas as pd
import numpy as np
import rich
import re

from tools import geodlparse, hwmaml_breastcancer_trainandtest
from sklearn import svm
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.model_selection import StratifiedKFold, train_test_split, cross_val_score
```

In [ ]:
```
# Download and parse data
gse = geodlparse('GSE7390')
gse_data = pd.concat(
    [ gsm.table.set_index('ID_REF')['VALUE'] for _,gsm in gse.gsms.items() ],
    axis=1
).set_axis([ x for x,_ in gse.gsms.items() ], axis=1, inplace=False)

# Retrieve sample groups (labels)
groups = gse.phenotype_data.filter(regex='e\\.rfs$', axis=1) \
    .replace({'0': 'No Replapse', '1': 'Relapse'}).sort_index()

# Select the 76 genes identified in Wang, 2005
with open('data/genelist.txt', 'r') as file:
    genelist = [re.match(r'^\d{6}\w+', x)[0] for x in file.readlines()]

gse_data = gse_data.filter(genelist, axis=0).T \
    .rename_axis('', axis=1) \
    .sort_index(axis=1)

# Define variables for ml
X = gse_data.values
X = StandardScaler().fit_transform(X)  # normalize data
y = groups['characteristics_ch1.14.e.rfs'].values
genes = gse_data.columns
```

```
Loading cached data...
```

In [ ]:
```
# Split the data into training (90%) and testing sets (10%)
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.9, random_state=69)

# Split the data into stratified folds and select the first partition
skf = StratifiedKFold(n_splits=4, random_state=69, shuffle=True)
train_idx, test_idx = list(skf.split(X_train, y_train))[0]
X_train, y_train = X_train[train_idx], y_train[train_idx]

# Fit the training data to the SVM
clf = svm.SVC(kernel='rbf')
clf.fit(X_train, y_train)
```

```
# Get model predictions
y_pred = clf.predict(X_test)

# Calculate and report model accuracy
accuracy = (y_pred == y_test).mean()
rich.print(f'The accuracy of the SVM model for a single fold is {accuracy*100:.2f}%')
```

```
The accuracy of the SVM model for a single fold is 60.00%
```

Write an evaluation function `hwmalml_breastcancer_trainandtest(X_train, y_train, X_test, y_test)` that trains an SVM using `X_train` and `y_train`, where `X_train` is the gene expression data for a subset of the samples, and `y_train` is a binary vector of class labels (indicating cancer relapse status) and calculates the **accuracy** on the test data (`X_test` and `y_test`).

The `hwmaml_breastcancer_trainandtest` function is defined in the `tools.py` module.

In [ ]:
```
accuracy = hwmaml_breastcancer_trainandtest(X_train, y_train, X_test, y_test)
rich.print(f'The accuracy of the SVM model for a single fold is {accuracy*100:.2f}%')
```

```
The accuracy of the SVM model for a single fold is 60.00%
```

Feature Selection

Perform forward selection of features (genes) that give the best prediction results (as measured by accuracy).

- Create a 10-fold cross-validation of all data samples
- Report the names of the genes that were selected to have the best accuracy

In [ ]:
```
# Initialize cross-validator
skf = StratifiedKFold(n_splits=10, random_state=69, shuffle=True)
# Initialize SVM classifier
clf = svm.SVC(kernel='rbf')
# Perform feature selection
sfs = SequentialFeatureSelector(clf, direction='forward', cv=skf, n_jobs=-1)
sfs.fit(X, y);

rich.print('Feature Selection resulted in the following genes:\n',
           ', '.join(genes[sfs.get_support()]), sep='')
```

```
Feature Selection resulted in the following genes:
200965_s_at, 201068_s_at, 201368_at, 201663_s_at, 201664_at, 202239_at, 202687_s_at,
203306_s_at, 203391_at, 204073_s_at, 204218_at, 205034_at, 205848_at, 208683_at, 209524_at,
209835_x_at, 210028_s_at, 211382_s_at, 211762_s_at, 212014_x_at, 212567_s_at, 214919_s_at,
216693_x_at, 217102_at, 217404_s_at, 217471_at, 217771_at, 217815_at, 218430_s_at,
218533_s_at, 218914_at, 219510_at, 219588_s_at, 219724_s_at, 220886_at, 221028_s_at,
221241_s_at, 221634_at
```

In [ ]:
```
# Using the list of genes selected, report the 10-fold cross-validation accuracy
# of the SVM model
accuracy = cross_val_score(clf, X[:,sfs.get_support()], y, cv=skf)
rich.print(f'The SVM accuracy is {accuracy.mean()*100:.3f}%')
```

```
The SVM accuracy is 73.842%
```