The journal of the irreproducible cancer research

Source code https://github.com/noskill/mlcancer

Each study was split: training set 0.9 of rows, and validation set 0.1 of rows.

Each validation set is perfectly balanced with equal count of zeros and ones in posOutcome. Balancing procedure splits each study data into positive and negative outcomes and shuffles. Validation set is then the first 0.1 rows for positive and negative tables. Thus the validation set has an even number or rows and is perfectly balanced by outcome.

This subset of treatment variables from **bcClinicalTable.csv** was uses:

tumor_size_cm_preTrt_preSurgery, tumor_size_cm_secondAxis_preTrt_preSurgery, preTrt_lymph_node_status, preTrt_totalLymphNodes, preTrt_numPosLymphNodes, hist_grade, nuclear_grade_preTrt, age, race, menopausal_status, surgery_type, intarvenous, intramuscular, oral, radiotherapyClass, chemotherapyClass, hormone_therapyClass, postmenopausal_only, immediate_biol_target, anthracycline, taxane, anti_estrogen, aromatase_inhibitor, estrogen_receptor_blocker, estrogen_receptor_blocker_and_stops_production, estrogen_receptor_blocker_and_eliminator, anti_HER2, tamoxifen, doxorubicin, epirubicin, docetaxel, capecitabine, fluorouracil, paclitaxel, cyclophosphamide, anastrozole, fulvestrant, gefitinib, trastuzumab, letrozole, chemotherapy, hormone_therapy, no_treatment, methotrexate, cetuximab, carboplatin, other, taxaneGeneral

Description of each variable is in **bcTabs.ods**

There are 4 experiments:

1) genes expression + treatment
2) just treatment alone
3) just genes expression
4) just averaged treatment from **bmc15mldata1.csv**(radio, chemo, surger)

Best results have been achieved with genes expression + treatment with svm classifier.

Metrics:

For validation set:
accuracy

recall
precision
f1
Confusion matrix
For train set:
f1
Confusion matrix

Algorithms:

SVM, nearest neighbour
For svm test results are average of 10 or 30 iterations to decrease variance.

SVM tests

Experiment1: genes expression + treatment 30 iterations

Train shape   (1985, 8880)
Validation shape (240, 8880)

model = svm.SVC(C=1, kernel='rbf', class_weight={1: 0.5})

```
recall: 0.5927777777777777
precision: 0.61585600531111
f1: 0.6034480753122881
confusion: [[75.5        44.5       ]
            [48.86 71.13333333]]
train_f1: 0.8639638708794571
train_confusion: [[ 667.56666667   51.43333333]
                  [ 264.03333333 1001.96666667]]
accuracy: 0.6109722222222224
```

Balanced by number of samples from each study in the train set

Train shape (4352, 8880)
Validation shape (240, 8880)

```
recall: 0.7816666666666665
precision: 0.5823507575921639
f1: 0.6672835225254414
confusion: [[52.73333333 67.26666667]
            [26.2        93.8       ]]
```

```
train_f1: 0.9719922245898863
train_confusion: [[1312.63333333   96.33333333]
                  [  69.26666667 2873.76666667]]
accuracy: 0.6105555555555554
```

```
model = svm.SVC(C=1, kernel='linear', class_weight={1: 0.5})
```

```
recall: 0.7
precision: 0.5384615384615384
f1: 0.608695652173913
confusion: [[48 72]
 [36 84]]
train_f1: 1.0
train_confusion: [[ 719    0]
 [   0 1266]]
accuracy: 0.55
```

Experiment2: treatment iterations 10

```
Train shape  (1985, 48)
Validation shape (240, 48)
```

```
model = svm.SVC(C=1, kernel='rbf', class_weight={1: 0.5})
```

```
recall: 0.35750000000000004
precision: 0.5022006496611948
f1: 0.41758949931093853
confusion: [[77.5 42.5]
            [77.1 42.9]]
train_f1: 0.5868362433627302
train_confusion: [[584.7 134.3]
                  [684.3 581.7]]
accuracy: 0.5016666666666667
```

```
model = svm.SVC(C=1, kernel='linear', class_weight={1: 0.5})
```

```
recall: 0.6333333333333333
```

```
precision: 0.5
f1: 0.5588235294117647
confusion: [[44 76]
 [44 76]]
train_f1: 0.8076616121308858
train_confusion: [[ 491  228]
 [ 254 1012]]
accuracy: 0.5
```

Experiment3: genes expression: 30 iterations

```
Train shape (1985, 8832)
Validation shape (240, 8832)
```

model = svm.SVC(C=1, kernel='rbf', class_weight={1: 0.5})

```
recall: 0.5772222222222222
precision: 0.6224067750131447
f1: 0.5983274390972145
confusion: [[77.86666667 42.13333333]
            [50.73333333 69.26666667]]
train_f1: 0.8713440414619649
train_confusion: [[ 676.4   42.6]
                  [ 255.7 1010.3]]
accuracy: 0.6130555555555554
```

model = svm.SVC(C=1, kernel='linear', class_weight={1: 0.5})

```
recall: 0.6333333333333333
precision: 0.5588235294117647
f1: 0.59375
confusion: [[60 60]
            [44 76]]
train_f1: 1.0
train_confusion: [[ 719    0]
                  [   0 1266]]
accuracy: 0.5666666666666667
```

Experiment4: genes expression + averaged: treatment 10 iterations

Train shape (1985, 8836)
Validation shape (240, 8836)

model = svm.SVC(C=1, kernel='rbf', class_weight={1: 0.5})

```
recall: 0.5716666666666667
precision: 0.6120916218310122
f1: 0.5903127976185885
confusion: [[76.5 43.5]
            [51.4 68.6]]
train_f1: 0.873382181836979
train_confusion: [[ 676.4   42.6]
                  [ 251.5 1014.5]]
accuracy: 0.6045833333333335
```

model = svm.SVC(C=1, kernel='linear', class_weight={1: 0.5})

```
recall: 0.575
precision: 0.5307692307692308
f1: 0.5519999999999999
confusion: [[59 61]
            [51 69]]
train_f1: 1.0
train_confusion: [[ 719    0]
                  [   0 1266]]
accuracy: 0.5333333333333333
```

Run separate SVM per study: 20 iterations

Experiment1: genes expression + treatment

Parameter **C** varies between studies

```
model = svm.SVC(C=C, kernel='rbf', class_weight={1: (1 - mean(train_labels))  / mean(train_labels)})
```

```
recall: 0.726313025210084
precision: 0.588925771359299
f1: 0.6186857808965436
confusion: [[3.68235294 3.37647059]
 [2.04705882 5.01176471]]
train_f1: 0.9158447825533841
train_confusion: [[38.84705882  3.44705882]
 [ 7.49411765 66.97647059]]
accuracy: 0.6052871148459383
```

Details per study:

study_17705_GPL96_JBI_Tissue_BC_Tamoxifen-bmc15

```
recall: 0.8166666666666668
precision: 0.5366089466089465
f1: 0.6427000450529862
confusion: [[1.85 4.15]
 [1.1  4.9 ]]
train_f1: 0.9738855263967965
train_confusion: [[12.  0.]
 [ 4. 75.]]
accuracy: 0.5625000000000001
```

study_25055_GPL96_MDACC_M-bmc15

```
recall: 0.6333333333333334
precision: 0.5863972832722831
f1: 0.602773836300152
confusion: [[6.7 5.3]
 [4.4 7.6]]
train_f1: 0.8166180061828161
train_confusion: [[ 29.   2.]
 [ 50. 116.]]
accuracy: 0.5958333333333334
```

study_17705_GPL96_MDACC_Tissue_BC_Tamoxifen-bmc15

```
recall: 0.76
precision: 0.5390386900681018
```

```
f1: 0.6268145371028666
confusion: [[3.45 6.55]
 [2.4  7.6 ]]
train_f1: 0.9550511571695411
train_confusion: [[ 41.05   1.95]
 [  9.55 122.45]]
accuracy: 0.5525
```

study_9893_GPL5049_all-bmc15

```
recall: 0.88125
precision: 0.6744055944055943
f1: 0.7608807558110964
confusion: [[4.5  3.5 ]
 [0.95 7.05]]
train_f1: 0.8895951820755619
train_confusion: [[29.95 14.05]
 [ 6.25 81.75]]
accuracy: 0.721875
```

study_16446_GPL570_all-bmc15

```
recall: 0.9166666666666666
precision: 0.5256078643578643
f1: 0.6654578754578754
confusion: [[1.  5. ]
 [0.5 5.5]]
train_f1: 0.9929932120849372
train_confusion: [[19.    0.  ]
 [ 1.15 81.85]]
accuracy: 0.5416666666666667
```

study_22358_GPL5325_all-bmc15

```
recall: 0.8071428571428573
precision: 0.7275974025974026
f1: 0.7463243782361431
confusion: [[4.5  2.5 ]
 [1.35 5.65]]
train_f1: 0.5353565147691456
train_confusion: [[7.145e+01 2.355e+01]
 [5.000e-02 1.295e+01]]
accuracy: 0.7249999999999999
```

study_22226_GPL1708_all-bmc15

```
recall: 0.8928571428571429
precision: 0.5616656954156953
f1: 0.6850180598555211
confusion: [[2.   5.  ]
 [0.75 6.25]]
train_f1: 0.9867179869262831
train_confusion: [[26.1  0.9]
 [ 1.4 85.6]]
accuracy: 0.5892857142857142
```

study_20194_GPL96_all-bmc15

```
recall: 0.4749999999999999
precision: 0.889375901875902
f1: 0.6115465392730839
confusion: [[13.15  0.85]
 [ 7.35  6.65]]
train_f1: 0.9921686746987952
train_confusion: [[191.35   0.65]
 [  0.    41.  ]]
accuracy: 0.7071428571428572
```

study_20181_GPL96_all-bmc15

```
recall: 0.3333333333333333
precision: 0.5708333333333333
f1: 0.3947619047619048
confusion: [[2.55 0.45]
 [2.   1.  ]]
train_f1: 0.7728088072320527
train_confusion: [[13.   0. ]
 [12.5 21.5]]
accuracy: 0.5916666666666667
```

study_19615_GPL570_all-bmc15

```
recall: 0.7416666666666666
precision: 0.5808333333333333
f1: 0.6338375839846427
confusion: [[2.65 3.35]
```

```
 [1.55 4.45]]
train_f1: 0.8759692001621934
train_confusion: [[ 7.75  0.25]
 [18.75 71.25]]
accuracy: 0.5916666666666668
```

study_1379_GPL1223_all-bmc15

```
recall: 0.85
precision: 0.6258333333333332
f1: 0.7102380952380951
confusion: [[1.4  1.6 ]
 [0.45 2.55]]
train_f1: 0.939796239696212
train_confusion: [[22.35  2.65]
 [ 0.95 28.05]]
accuracy: 0.6583333333333333
```

study_25065_GPL96_USO-bmc15

```
recall: 0.9833333333333332
precision: 0.495
f1: 0.6583333333333332
confusion: [[0.   3.  ]
 [0.05 2.95]]
train_f1: 0.9970017636684304
train_confusion: [[ 6.8   0.2 ]
 [ 0.05 40.95]]
accuracy: 0.4916666666666666
```

study_25065_GPL96_MDACC-bmc15

```
recall: 0.8375
precision: 0.5242857142857142
f1: 0.6416161616161615
confusion: [[0.95 3.05]
 [0.65 3.35]]
train_f1: 0.9678553408050872
train_confusion: [[ 7.8  0.2]
 [ 3.2 51.8]]
accuracy: 0.5375
```

study_32646_GPL570_all-bmc15

```
recall: 0.025
precision: 0.15
f1: 0.04285714285714286
confusion: [[6.   0.  ]
 [5.85 0.15]]
train_f1: 1.0
train_confusion: [[83.  0.]
 [ 0. 20.]]
accuracy: 0.5125
```

study_16391_GPL570_all-bmc15

```
recall: 0.7333333333333333
precision: 0.615
f1: 0.6372619047619047
confusion: [[1.4 1.6]
 [0.8 2.2]]
train_f1: 0.9689124767761179
train_confusion: [[ 6.95  0.05]
 [ 2.   33.  ]]
accuracy: 0.6
```

study_2034_GPL96_all-bmc15

```
recall: 0.7633333333333332
precision: 0.6171087115069314
f1: 0.6795637507245585
confusion: [[ 7.9   7.1 ]
 [ 3.55 11.45]]
train_f1: 0.9333807650739516
train_confusion: [[ 85.8    6.2 ]
 [ 15.05 148.95]]
accuracy: 0.6449999999999999
```

study_12093_GPL96_all-bmc15

```
recall: 0.7571428571428571
precision: 0.5412247474747476
f1: 0.6176281389748882
confusion: [[2.55 4.45]
 [1.7  5.3 ]]
```

```
train_f1: 0.9551318009902966
train_confusion: [[12.9  0.1]
 [ 9.1 99.9]]
accuracy: 0.5607142857142857
```

Nearest neighbour

Experiment1: genes expression + treatment

```
(1985, 8880)
(240, 8880)

recall: 0.7416666666666667
precision: 0.5393939393939394
f1: 0.6245614035087719
confusion: [[44 76]
            [31 89]]
train_f1: 1.0
train_confusion: [[ 719    0]
                  [   0 1266]]
accuracy: 0.5541666666666667
```

Experiment2: treatment

```
(1985, 48)
(240, 48)

recall: 0.7
precision: 0.5121951219512195
f1: 0.5915492957746479
confusion: [[40 80]
             [36 84]]
train_f1: 0.9444444444444445
train_confusion: [[ 598  121]
                  [  25 1241]]
accuracy: 0.5166666666666667
```

Experiment3: genes expression

```
(1985, 8832)
(240, 8832)

recall: 0.7583333333333333
precision: 0.5352941176470588
f1: 0.6275862068965518
confusion: [[41 79]
            [29 91]]
train_f1: 1.0
train_confusion: [[ 719    0]
                 [   0 1266]]
accuracy: 0.55
```

Experiment4: genes expression + averaged treatment

```
(1985, 8836)
(240, 8836)

recall: 0.7666666666666667
precision: 0.5411764705882353
f1: 0.6344827586206897
confusion: [[42 78]
            [28 92]]
train_f1: 1.0
train_confusion: [[ 719    0]
                 [   0 1266]]
accuracy: 0.5583333333333333
```

To do: moses, autoencoder for genes data binarisation, logistic regression, bagging classifier, naive bayes variants