Overview

This project applies logistic regression (LR) to rank SNPs from several PGS datasets as predictors of Parkinson's disease.

Click on each dataset link to explore the details.

Cooper 142 SNPs set

Preparation

Import required packages.

```
import os, sys, warnings
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import GridSearchCV
from sklearn.exceptions import ConvergenceWarning
```

Read input matrix with genotypes

The matrix contains the genotypes from AMP-PD/MGRB dataset for 140 SNPs.

```
table = pd.read_csv("data/matrix.txt", sep="\t")
table
```

	participant_id	phenotype	cohort	gender	inv_genotype	rs2275579	rs144115304	rs115581042	rs79531911	rs138844738	...	rs10448130	rs3428
0	SY-NIH_INVAA791MKCET	1	STEADY-PD3	M	NI	0	0	0	0	0	...	2	
1	SY-NIH_INVEP886EEYYL	1	STEADY-PD3	M	NI	0	0	0	0	0	...	0	
2	SY-NIH_INVFM717GWDX4	1	STEADY-PD3	F	NI	0	0	0	0	0	...	0	
3	SY-NIH_INVNN611MKKN9	1	STEADY-PD3	M	NI	0	0	0	0	0	...	1	
4	SY-NIH_INVRB171EXGUK	1	STEADY-PD3	M	II	0	1	1	1	0	...	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3107	BABQX	0	MGRB	M	II	0	0	0	0	0	...	1	
3108	BABRB	0	MGRB	F	II	0	0	0	0	0	...	0	
3109	BABRE	0	MGRB	M	NI	0	0	0	0	0	...	1	
3110	ZAAAB	0	MGRB	M	NI	0	0	0	0	0	...	2	
3111	AABUO	0	MGRB	F	NN	0	0	0	0	0	...	1	

3112 rows × 145 columns

Distribution of data

Distribution by phenotype

(0=Control, 1=Case)

```
table.groupby('phenotype')['participant_id'].nunique()
```

```
phenotype
0    1556
1    1556
Name: participant_id, dtype: int64
```

Distribution by gender/phenotype

```
table.groupby(['gender', 'phenotype'])['participant_id'].nunique()
```

```
gender  phenotype
F       0            567
        1            567
M       0            989
        1            989
Name: participant_id, dtype: int64
```

Distribution by gender/phenotype/inv8_001 genotype

```
table.groupby(['gender', 'phenotype', 'inv_genotype'])['participant_id'].nunique()
```

```
gender  phenotype  inv_genotype
F       0          II              195
                   NI              259
                   NN              113
        1          II              175
                   NI              270
                   NN              122
M       0          II              318
                   NI              480
                   NN              191
        1          II              296
                   NI              477
                   NN              216
Name: participant_id, dtype: int64
```

All participants

Logistic regression model

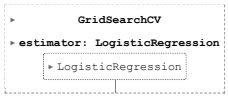
```
X = table[table.columns[5:]]
Y = table['phenotype']
lr = LogisticRegression(random_state=42, solver='saga', n_jobs=-1,
penalty='elasticnet')
```

Grid search for 3 hyperparameters

```
# parameters = {'C': [0.005, 0.01, 0.02, 0.05, 0.1, 0.5, 1, 10, 20, 30],
#               'max_iter': [10, 25, 50, 75, 100, 150, 200, 400, 800, 1600],
#               'l1_ratio': [1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]}

parameters = {'C': [0.01, 0.02, 0.05],
              'max_iter': [10, 25, 50],
              'l1_ratio': [1, 0.9, 0.8]}

grid_lr = GridSearchCV(lr, parameters, verbose=False, scoring='roc_auc',
n_jobs=-1, cv=10)
if not sys.warnoptions:
    warnings.simplefilter("ignore", category=ConvergenceWarning)
    os.environ["PYTHONWARNINGS"] = "ignore" # Also affect subprocesses
grid_lr.fit(X, Y)
```



Best estimator

```
best_lr = grid_lr.best_estimator_

max_auc_score = roc_auc_score(Y, best_lr.predict_proba(X)[:, 1])
coefs = best_lr.coef_[0, :]
num_coef = np.sum(coefs != 0)
X_header = np.array(X.columns)

data_array = np.vstack((X_header, coefs))
model_coefs = pd.DataFrame(data=data_array.T, columns=['SNP', 'Coefficient'])
print(f'Max AUC score:{max_auc_score}\n')
print(f'Non-zero coefficients: {num_coef}\n')
print(f'Best estimator: {grid_lr.best_estimator_}')
print(f'Scorer: {grid_lr.scorer_}')
print(f'Best params: {grid_lr.best_params_}')
print(f'Best score: {grid_lr.best_score_}\n')
# opt.lengthMenu = [50, 100, 200]
model_coefs[model_coefs['Coefficient'] != 0 ]
```

```
Max AUC score:0.5551641873897212

Non-zero coefficients: 2

Best estimator: LogisticRegression(C=0.02, l1_ratio=1, max_iter=25, n_jobs=-1,
                   penalty='elasticnet', random_state=42, solver='saga')
Scorer: make_scorer(roc_auc_score, needs_threshold=True)
Best params: {'C': 0.02, 'l1_ratio': 1, 'max_iter': 25}
Best score: 0.5492252417764205
```

	SNP	Coefficient
82	rs11248057	0.055163
87	rs3806760	0.098122

By Razvan Sultana