CSE5370: Bioinformatics

Prof: Dr.Jacob luber

September 19, 2022

Genome Wide Association Studies

1 Generating Unique DataSet

"I have used the datasetGeneratorV2.py to generate the dataset.", using python3 datasetGenerator.py –ID [1001958998]. datasetgeneratorV2.py. My result was a csv file of 1000 records, performed manipulations by importing libraries and storing the data using Numpy and Pandas.

```
import numpy as np
import pandas as pd
from scipy.stats import hypergeom
import scipy.stats as stats

#Part1
mydata = open("1001958998.csv","r")
dataByLine = csv.reader(mydata,delimiter = ";")

snps = []
temp = True
for line in dataByLine:
    if temp:
        temp = False
        continue

    r = line[0].split(",")

    for x in range(1,5):
        r[x] = int(r[x])

    snps.append(r)

print(snps)
```

2 Fisher's Exact Test

The SNP's whose value is known is be less than or equal to 5*10^-8. My null hypothesis indicates that all SNP's are independent for which I've opted for the two-sided test and obtained a total of 338 SNP's.

```
outputFile = open('results.csv', 'w+')
outputHook = csv.writer(outputFile)

plot = [[],[]]
x = 1
```

```
significant_pValue = 0
rectified_pValue = 0

for snp in dataStore:
    table = np.array([[snp[1], snp[2]], [snp[3], snp[4]]])

    operation , pValue = fisher_exact(table, alternative= "two-sided")
```

2.1 Bonferroni-corrected p-value

Implemented the Bonferroni-Correct-p value test and observed negative significant values for the initially generated P value. The Bonferroni correction is an adjustment made to P values when several dependent or independent statistical tests are being performed simultaneously on a single data set. The Bonferroni-test outputted a total of 240 significant SNP's

```
eff_pValue = 5 * (10 ** (-8))
    rectified_pValue = eff_pValue / 1000

    if pValue <= eff_pValue:
        significant_pValue += 1
        if pValue <= significant_pValue:
            rectified_pValue  += 1

    outputHook.writerow([pValue, pValue <= eff_pValue, pValue <= rectified_pValue])

    plot[0].append(x)
    x += 1
    plot[1].append(-np.log10(pValue))

outputFile.close()
```

2.2 Generating the Manhattan Plot

Generated the Manhattan plot using the log10 (p values) with the original and corrected p-value threshold. The SNP vales are along the x axis, with the first line representing the p-values of SNP'S scattered and the second line representing the Bonferreti-corrected p values. The SNP'S that are observed above the line, are expected cause diseases. See the code for Figure **??**

```
#PART4
plt.axhline(y=-np.log10(5*(10**(-11))), color='r', linestyle='-')
plt.axhline(y=-np.log10(5*(10**(-8))), color='r', linestyle='-')
plt.scatter(plot[0], plot[1])

plt.xlabel("SNP locus")
plt.ylabel("-log10(p_values)")

plt.show()
```

See the code for Figure **??**

2.3 References

Used geek for geeks, Stack over flow, Official python documentation, and classmate collaboration.

2.4 Difficulty Adjustment

The assignment took me over 10 hours to complete, I had/still have difficulty understanding the working of the biological part in the first context of the assignment. Also, I had taken time with the statistics part, I got stuck while trying to generate the alternate values for the fisher's test, the code was constantly breaking. I personally feel a detailed explanation of the assignments should be provided or the difficulty could be adjusted a bit. I am looking forward for the solution of this assignment. Thank you!