Singh2015_solution

January 23, 2017

1 Solution of Singh *et al.* 2015

1.1 Compute the mean `RecombinantFraction` for each Drosophila `Line` and `InfectionStatus`. Print the results like:

```
Line 45 Average Recombination Rate: W :  0.187 I :  0.191
```

To read the data, we're going to use the `csv` module. First, we need to import it:

```
In [1]: import csv
```

Then, we need to go through all the rows in the file, and for each add the `RecombinantFraction` to the right genetic `Line` and `InfectionStatus`. To do so, we need to choose a data structure. Here we use a dictionary, where the keys are given by `Line`, and each value of the dictionary is another dictionary where the keys `W` and `I` index lists of `RecombinantFraction`.

```
In [2]: my_data = {}
```

```
In [3]: with open('../data/Singh2015_data.csv') as csvfile:
            reader = csv.DictReader(csvfile)
            for row in reader:
                my_line = row['Line']
                my_status = row['InfectionStatus']
                my_recomb = float(row['RecombinantFraction'])
                # Test by printing the values
                print(my_line, my_status, my_recomb)
                # just print the first row
                break
```

```
21 I 0.1826923077
```

Now we need to perform operations for each row. First, we're going to check whether `my_data` already contains the `Line` for that row. If not, we'll create the key-value in the dictionary. Then, we're going to add the value to the list.

```
In [4]: with open('../data/Singh2015_data.csv') as csvfile:
            reader = csv.DictReader(csvfile)
            for row in reader:
```

```
        my_line = row['Line']
        my_status = row['InfectionStatus']
        my_recomb = float(row['RecombinantFraction'])
        # if my_line is not present in the dictionary:
        if my_line not in my_data:
            # create and initialize with a dictionary containing
            # two empty lists
            my_data[my_line] = {'W': [], 'I': []}
        # Now insert the value in the right list
        my_data[my_line][my_status].append(my_recomb)
```

Now we should have the data organized in a nice structure:

```
In [5]: my_data
```

```
Out[5]: {'21': {'I': [0.1826923077,
           0.1850393701,
           0.1856540084,
           0.1866666667,
           0.1904761905,
           0.1958762887,
           0.2180094787,
           0.2534246575],
          'W': [0.1288343558,
           0.163141994,
           0.1674208145,
           0.1746478873,
           0.175,
           0.1779661017,
           0.191588785,
           0.1961722488,
           0.2026578073,
           0.2032258065]},
         '40': {'I': [0.1573426573,
           0.1614173228,
           0.1666666667,
           0.1693989071,
           0.1740890688,
           0.1779141104,
           0.1878980892,
           0.2110552764,
           0.2153846154],
          'W': [0.125,
           0.156424581,
           0.1564885496,
           0.1595744681,
           0.1602209945,
           0.1651376147,
```

```
  0.1694915254,
  0.1700404858,
  0.1710526316,
  0.180952381,
  0.1828793774,
  0.1888888889,
  0.1892857143,
  0.2123287671,
  0.2247706422,
  0.2340425532]},
'45': {'I': [0.1666666667,
  0.1736111111,
  0.1838565022,
  0.1862068966,
  0.1873015873,
  0.1875,
  0.188976378,
  0.1981707317,
  0.1993355482,
  0.2068965517,
  0.2077922078,
  0.2080745342],
 'W': [0.1481481481,
  0.1625,
  0.175862069,
  0.1859504132,
  0.1906779661,
  0.2007722008,
  0.2032967033,
  0.2033195021,
  0.213740458]},
'73': {'I': [0.1666666667,
  0.1812297735,
  0.1818181818,
  0.1850746269,
  0.2109090909,
  0.2179487179,
  0.2183098592,
  0.2339449541],
 'W': [0.1551724138,
  0.1573033708,
  0.1653543307,
  0.1678321678,
  0.1744680851,
  0.1802721088,
  0.1944444444,
  0.1952861953,
  0.1956521739,
```

```
            0.2]}}
```

Time to calculate the means and print the results:

```
In [6]: for line in my_data:
            print('Line', line, 'Average Recombination Rate:')
            # extract the relevant data
            my_subset = my_data[line]
            for status in ['W', 'I']:
                print(status, ':', end = '') # to prevent new line
                my_mean = sum(my_subset[status])
                my_num_elements = len(my_subset[status])
                my_mean = my_mean / my_num_elements
                print(' ', round(my_mean, 3))
            print('') # to separate the lines

Line 45 Average Recombination Rate:
W :  0.187
I :  0.191

Line 21 Average Recombination Rate:
W :  0.178
I :  0.2

Line 40 Average Recombination Rate:
W :  0.178
I :  0.18

Line 73 Average Recombination Rate:
W :  0.179
I :  0.199
```