In [1]:

```
import sys
import pandas as pd
import numpy as np
from tqdm import tqdm
import matplotlib.pyplot as plt
```

In [2]:

```
def parse_chromosome(pandas_series):
    return pandas_series.apply(lambda x: int(x[3:]))
```

In [3]:

```python
def create_bed(data, window):

    cpg_islands = []
    shores = []
    shelves = []
    seas = []

    prev_start = data.iloc[0]['start']
    prev_stop = data.iloc[0]['stop']
    prev_chromosome = data.iloc[0]['chromosome']
    prev_chromosome_len = data.iloc[0]['length']

    cpg_islands.append((prev_chromosome, prev_start, prev_stop))
    if prev_start > (2*window):
        seas.append((prev_chromosome, 0, prev_start - 2*window))
        shelves.append((prev_chromosome, prev_start - 2*window, prev_start - window))
        shores.append((prev_chromosome, prev_start - window, prev_start))
    elif prev_start > (window):
        shelves.append((prev_chromosome, 0, prev_start - window))
        shores.append((prev_chromosome, prev_start - window, prev_start))
    elif prev_start < (window):
        shores.append((prev_chromosome, 0, prev_start))
    else:
        sys.exit("Something's wrong")


    for i in tqdm(range(1, data.shape[0]), desc = 'CPG Islands'):
        current_start = data.iloc[i]['start']
        current_stop = data.iloc[i]['stop']
        current_chromosome = data.iloc[i]['chromosome']
        current_chromosome_len = data.iloc[i]['length']

        if prev_chromosome == current_chromosome:
            if current_start - prev_stop > (4*window):
                shores.append((current_chromosome, prev_stop, prev_stop + window))
                shelves.append((current_chromosome, prev_stop + window, prev_stop + 2*window))
                seas.append((current_chromosome, prev_stop + 2*window, current_start - 2*window))
                shelves.append((current_chromosome, current_start - 2*window, current_start - window))
                shores.append((current_chromosome, current_start - window, current_start))
            elif current_start - prev_stop > (2*window):
                shores.append((current_chromosome, prev_stop, prev_stop + window))
                shelves.append((current_chromosome, prev_stop + window, current_start - window))
                shores.append((current_chromosome, current_start - window, current_start))
            elif current_start - prev_stop > (window):
                shores.append((current_chromosome, prev_stop, current_start))
            elif current_start - prev_stop <= (window):
                shores.append((current_chromosome, prev_stop, current_start))
            else:
                sys.exit("Something's wrong")
        elif prev_chromosome != current_chromosome:
            if prev_chromosome_len - prev_stop > (2*window):
                shores.append((prev_chromosome, prev_stop, prev_stop + window))
                shelves.append((prev_chromosome, prev_stop + window, prev_stop + 2*window))
                seas.append((prev_chromosome, prev_stop + 2*window, prev_chromosome_len))
            elif prev_chromosome_len - prev_stop > (window):
                shores.append((prev_chromosome, prev_stop, prev_stop + window))
                shelves.append((prev_chromosome, prev_stop + window, prev_chromosome_len))
            elif prev_chromosome_len - prev_stop <= (window):
                shores.append((prev_chromosome, prev_stop, prev_chromosome_len))
            else:
                sys.exit("Something's wrong")

            if current_start > (2*window):
                seas.append((current_chromosome, 0, current_start - 2*window))
                shelves.append((current_chromosome, current_start - 2*window, current_start - window))
                shores.append((current_chromosome, current_start - window, current_start))
            elif current_start > (window):
                shelves.append((current_chromosome, 0, current_start - window))
                shores.append((current_chromosome, current_start - window, current_start))
            elif current_start <= (window):
                shores.append((current_chromosome, 0, current_start))
            else:
                sys.exit("Something's wrong")

        prev_start = current_start
        prev_stop = current_stop
        prev_chromosome = current_chromosome
        prev_chromosome_len = current_chromosome_len
        cpg_islands.append((current_chromosome, current_start, current_stop))

    if prev_chromosome_len - prev_stop > (2*window):
        shores.append((prev_chromosome, prev_stop, prev_stop + window))
        shelves.append((prev_chromosome, prev_stop + window, prev_stop + 2*window))
        seas.append((prev_chromosome, prev_stop + 2*window, prev_chromosome_len))
    elif prev_chromosome_len - prev_stop > (window):
        shores.append((prev_chromosome, prev_stop, prev_stop + window))
        shelves.append((prev_chromosome, prev_stop + window, prev_chromosome_len))
    elif prev_chromosome_len - prev_stop <= (window):
        shores.append((prev_chromosome, prev_stop, prev_chromosome_len))
    else:
        sys.exit("Something's wrong")

    return cpg_islands, shelves, shores, seas
```

In [4]:

```
def save_bed_format(filename, data):
    with open(filename+'.bed', 'w') as f:
        for row in data:
            f.write("%s\t%s\t%s\n" % row)
```

In [5]:

```
def calculate_hits(cpg_islands, shores, shelves, seas, dna_methylation):
    """
    All data must be in tuples
    (chromosome, start, stop)

    """
    # Prepare data
    cpg_islands_df = pd.DataFrame(cpg_islands, columns = ['chromosome','start',
'stop'])
    cpg_islands_df['region'] = 'cpg_island'
    shores_df = pd.DataFrame(shores, columns = ['chromosome','start','stop'])
    shores_df['region'] = 'shore'
    shelves_df = pd.DataFrame(shelves, columns = ['chromosome','start','stop'])
    shelves_df['region'] = 'shelve'
    seas_df = pd.DataFrame(seas, columns = ['chromosome','start','stop'])
    seas_df['region'] = 'sea'
    all_data = cpg_islands_df.append(shores_df, ignore_index=True).append(shelve
s_df, ignore_index=True).append(seas_df, ignore_index=True)
    all_data = all_data.sort_values(['chromosome','start'])

    regions = []
    for row in tqdm(dna_methylation, desc='DNA Methylation'):
        middle = row[1] + (row[2] - row[1])/2
        all_data_rows = all_data[(all_data['chromosome'] == row[0]) & (all_data[
'start'] <= middle) & (all_data['stop'] >= middle)]

        if all_data_rows.shape[0] > 1:
            all_data_rows = all_data_rows.iloc[[-1]]
        if all_data_rows.shape[0] == 0:
            print('no match')
            print(row)

        regions.append(all_data_rows['region'].values)
    return regions
```

In [7]:

```
cpg_islands = pd.read_csv('cpgIslandExt.txt', sep = '\t', header = None).iloc[:,
1:4]
cpg_islands = cpg_islands.rename(columns = {1:'chromosome',2:'start',3:'stop'})
chromosomes = ['chr1','chr2','chr3','chr4','chr5','chr6','chr7','chr8','chr9','c
hr10','chr11','chr12','chr13','chr14',
               'chr15','chr16','chr17','chr18','chr19','chr20','chr21','chr22']
cpg_islands = cpg_islands[cpg_islands['chromosome'].isin(chromosomes)]
cpg_islands['chromosome'] = parse_chromosome(cpg_islands['chromosome'])
cpg_islands = cpg_islands.sort_values(['chromosome', 'start'])
```

In [8]:

```
chromosomes_lengths = pd.read_csv('hg19.chrom.sizes.txt', sep = '\t', header = None, names = ['chromosome','length'])
chromosomes_lengths = chromosomes_lengths[chromosomes_lengths['chromosome'].isin(chromosomes)]
chromosomes_lengths['chromosome'] = parse_chromosome(chromosomes_lengths['chromosome'])
data = cpg_islands.merge(chromosomes_lengths)
```

In [9]:

```
chromosomes_lengths.head()
data.head()
```

Out[9]:

	chromosome	start	stop	length
0	1	28735	29810	249250621
1	1	135124	135563	249250621
2	1	327790	328229	249250621
3	1	437151	438164	249250621
4	1	449273	450544	249250621

In [10]:

```
dna_methylation = []
with open("data.bed")as f:
    for line in f:
        row = line.strip().split()
        if row[0] in chromosomes:
            chromosome = int(row[0][3:])
            start = int(row[1])
            stop = int(row[2])
            dna_methylation.append((chromosome, start, stop))
dna_methylation[0:5]
```

Out[10]:

```
[(16, 53468112, 53468162),
 (3, 37459206, 37459256),
 (3, 171916037, 171916087),
 (1, 91194674, 91194724),
 (8, 42263294, 42263344)]
```

In [11]:

```
cpg_islands, shelves, shores, seas = create_bed(data, 2000)
```

```
CPG Islands: 100%|██████████| 26640/26640 [00:17<00:00, 1542.15it/s]
```

In [12]:

```
save_bed_format('cpg_islands', cpg_islands)
save_bed_format('shelves', shelves)
save_bed_format('shores', shores)
save_bed_format('seas', seas)
```

In [13]:

```
regions = calculate_hits(cpg_islands, shores, shelves, seas, dna_methylation)
```

```
DNA Methylation: 100%|██████████| 470870/470870 [24:16<00:00, 323.20
it/s]
```

In [14]:

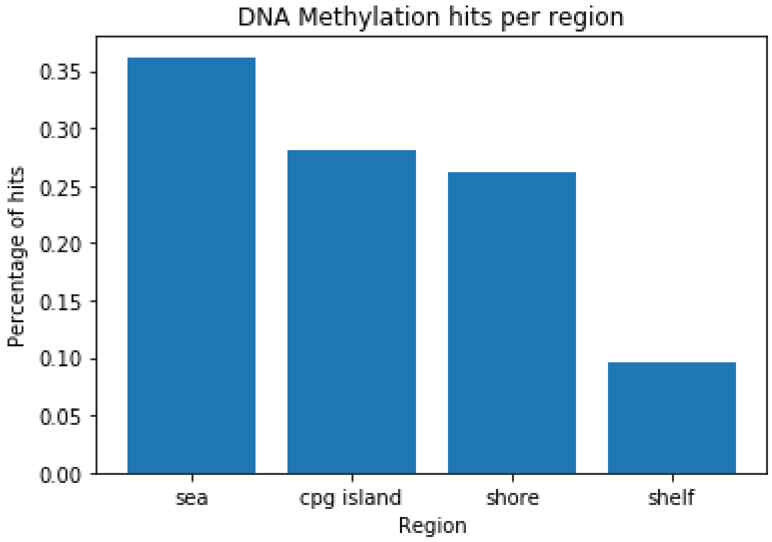
```
hits = pd.Series(regions).value_counts(normalize=True)
hits
```

Out[14]:

```
[sea]           0.361896
[cpg_island]    0.280532
[shore]         0.261034
[shelve]        0.096538
dtype: float64
```

In [15]:

```
fig, ax = plt.subplots()
plt.bar(['sea','cpg island','shore','shelf'],hits)
plt.title('DNA Methylation hits per region')
plt.ylabel('Percentage of hits')
plt.xlabel('Region')
plt.show()
```



In [16]:

```
np.sum(pd.Series(regions).value_counts())
```

Out[16]:

```
470870
```

In [ ]: