Step 1:

The following function calculate the amount of every single one pair of nucleotides. And return a dictionary with the name of pair and the amount of this pair in sequence. For this exercise I use the file Staaur.fa.

```
import regex as re
import numpy as np

def kmers(genomefile, k):

    file = open(genomefile, 'r')
    seq = ""
    kmer_table = {}
    count = 0
    for line in file:
        count +=1
        if (count > 1) :
            length=len(line)
            seq=seq+line[0:length-1]

    file.close()

    seq = re.sub("[^AGCT]", "", seq)

    for i in range(len(seq)-k):
        DNA=seq[i:i+k]
        if DNA not in kmer_table.keys():
            kmer_table[DNA]=1
        else:
            kmer_table[DNA]+=1

    kmer_table = {k: v for k, v in kmer_table.items()}

    return(kmer_table)

amount_pairs = kmers('Staaur.fa', 2)
```

The amount of pairs equals to:

{'CT': 16, 'TA': 19, 'AG': 12, 'GA': 25, 'AC': 63, 'TG': 12, 'GG': 14, 'GC': 16, 'CC': 129, 'CG': 25, 'CA': 53, 'AA': 66, 'AT': 21, 'TT': 19, 'GT': 8, 'TC': 14}

Step 2:

I input the values of amount in a array with the name ‘freq_seq’.

```
freq_seq = np.array(list(amount_pairs.values()))
```

Equals to [16 19 12 25 63 12 14 16 129 25 53 66 21 19   8 14]

Step 3:

Then I calculate the mean of amount of pairs (μ).

mean_freq = np.mean(freq_seq)

Equals to 32.0

Step 4:

I calculate the euclidean distance based on the following formula:

$$Euclidian\ Distance = \sqrt{\sum (x_i - \mu)^2}$$

*euclidean_distance = np.sqrt(np.sum((freq_seq – mean_freq)**2))*

Equals to 122.71919165313956

Step 5:

I apply the Z-transformation based on the following formula:

$$Z\ Distance = (Euclidian\ Distance - \mu)/\sigma$$, σ is the variance of Normal Distribution

z_distance = (euclidean_distance - mean_freq) / np.std(freq_seq)

Equals to 2.9569683577953607