

# 西安电子科技大学

## 安全前沿讨论班（I） 课程实验报告

实验名称 DNA 翻译和凯撒密码

学 院 网络与信息安全学院 班 级 1618019

姓 名 曹寅峰 学 号 16020610025

实验日期 2019 年 5 月 14 日

指导教师评语：

指导教师：

        年        月        日

## 实验报告内容基本要求及参考格式

### 一、实验目的

应用已经掌握的 Python 程序设计语言的相关知识，进行问题研究。1) 通过分析 DNA 翻译的过程，对所设计的问题进行编程，从而达到对所学知识的练习和巩固；2) 理解并实现凯撒加密过程。

### 二、实验环境

Jupyter notebook python3.7

### 三、实验基本原理及步骤（或方案设计及理论计算）

```
# create a function named read_seq with a parameter inputfile;
# the function reads the content of inputfile and stores to a variable seq;
# replace all '\n' and '\r' in seq to '';
# then return seq.
def read_seq(inputfile):
    """Reads and returns the input sequence with special characters removed."""
    # your code here
    f=open(inputfile,'r')
    seq=f.read()
    f.close()
    seq=seq.replace('\n','')
    seq=seq.replace('\r','')
    return seq
```

首先用 open 函数读入文件描述符，用 read 读入字符串，关闭文件后用 replace 替换回车与换行符，返回处理后的字符串

```
def translate(seq):
    """Translate a string containing a nucleotide sequence
    into a string containing the corresponding sequence of
    amino acids. Nucleotides are translated in triplets using
    the table dictionary; each amino acid is encoded with
    a string of length 1."""
    table = {
        'ATA':'I', 'ATC':'I', 'ATT':'I', 'ATG':'M',
        'ACA':'T', 'ACC':'T', 'ACG':'T', 'ACT':'T',
        'AAC':'N', 'AAT':'N', 'AAA':'K', 'AAG':'K',
        'AGC':'S', 'AGT':'S', 'AGA':'R', 'AGG':'R',
        'CTA':'L', 'CTC':'L', 'CTG':'L', 'CTT':'L',
        'CCA':'P', 'CCC':'P', 'CCG':'P', 'CCT':'P',
        'CAC':'H', 'CAT':'H', 'CAA':'Q', 'CAG':'Q',
        'CGA':'R', 'CGC':'R', 'CGG':'R', 'CGT':'R',
        'GTA':'V', 'GTC':'V', 'GTG':'V', 'GTT':'V',
        'GCA':'A', 'GCC':'A', 'GCG':'A', 'GCT':'A',
        'GAC':'D', 'GAT':'D', 'GAA':'E', 'GAG':'E',
        'GGA':'G', 'GGC':'G', 'GGG':'G', 'GGT':'G',
        'TCA':'S', 'TCC':'S', 'TCG':'S', 'TCT':'S',
        'TTC':'F', 'TTT':'F', 'TTA':'L', 'TTG':'L',
        'TAC':'Y', 'TAT':'Y', 'TAA':'_', 'TAG':'_',
        'TGC':'C', 'TGT':'C', 'TGA':'_', 'TGG':'W'
    }
    # check that the sequence length is divisible by 3
    # loop over the sequence, extract a single codon
    # loop up the codon and store the result
    protein = ""
    genlist=[]
    if len(seq)%3!=0:
        print("not divisible by 3!\n")#如果不能被3整除，则报错返回
        return 0
    for i in range(0,len(seq),3):#每三个为一组加入genlist
        genlist.append(seq[i:i+3])
    res=[table[x] if x in table else x for x in genlist]#根据table与genlist产生翻译后的序列 (list)
    return ''.join(res)#把list转化为string
```

然后进行基因序列的翻译。首先判断基因总数是否为 3 的倍数，如果不符合则返

回，其次每三个基因作为一个元素加入 genlist，然后根据 table 进行翻译，最后别忘了把 list 转化为 string。

对于去掉最后一个结束子，可以删除 seq 的后三个元素，也可以删除 res 的最后一个元素，效果相投。

```
# loop up the codon and store the result
protein = ""
genlist=[]
if len(seq)%3!=0:
    print("not divisible by 3!\n")
    return 0
seq=seq[0:-3]
for i in range(0,len(seq),3):
    genlist.append(seq[i:i+3])
res=[table[x] if x in table else x for x in genlist]
return ''.join(res)

# your code here
translate(dna[20:938])
```

```
# loop up the codon and store the result
protein = ""
genlist=[]
if len(seq)%3!=0:
    print("not divisible by 3!\n")
    return 0
for i in range(0,len(seq),3):
    genlist.append(seq[i:i+3])
res=[table[x] if x in table else x for x in genlist]
return ''.join(res[0:-1])
# your code here
translate(dna[20:938])
```

凯撒加密

```
# create `letters` here!
letters={}
for i in range(len(alphabet)):
    letters[i]=alphabet[i]
```

首先根据数字创建数字到字母的翻译字典

```
def encode_dict(encryption_key):
    encoding={}
    alphabet = string.ascii_lowercase + " "
    for i in range(len(alphabet)):  
        encoding[alphabet[i]]=(i+encryption_key)%27
    return encoding
```

其次创建字母到加密结果的字典。利用 alphabet[i]作为键，i+key(%27)为值

```
def caesar(message, encryption_key):
    # return the encoded message as a single string!
    encoded_message = ''
    # use the function in Step 2 to get the encoding dictionary
    encoding = encode_dict(encryption_key)
    for char in message:
        encoded_message+=letters[encoding[char]]
    return encoded_message
    # your code is here
    # for each letter in message, get the encoded letter
```

最后先将明文替换成加密后的数字，再将数字替换为字母表中的字母，生成密文。

#### 四、实验结果分析及回答问题（或测试环境及测试结果）

##### 1. dna 翻译结果

```
?j: dna = read_seq("dna.txt")
    # the locations of the gene where the coding sequence starts and ends 21-938
    translate(dna[20:938])

!j: 'MSTHDTSLKTTEEVAFAQIILLQCQFGVGTANVFLFVYNFSPISTGSKQRPRQVILRHMAVANALTFLTIFPNNMMTFAPIIPQTDLKCKLEFFTRLVARSTNLCSTCVLSIHQFV
TLVPVNSGKILRASVTNMAASYSCYSCWFFSVLNNIYIPIKVTGPQLTDNNNSKSKLFCSTSDFSVGIVFLRFAHDATFMSIMVWTSVSMVLLHHRHCQRMQYIFTLNQDPRGQAE
TTATHITILMLVVTFGFYLLSLICIIFYTYFIYSHSLRHCNDILVSGFPTISPLLLTFRDPKGPSCSVFFNC_'

!j: prt = read_seq("protein.txt")
    prt

!j: 'MSTHDTSLKTTEEVAFAQIILLQCQFGVGTANVFLFVYNFSPISTGSKQRPRQVILRHMAVANALTFLTIFPNNMMTFAPIIPQTDLKCKLEFFTRLVARSTNLCSTCVLSIHQFV
TLVPVNSGKILRASVTNMAASYSCYSCWFFSVLNNIYIPIKVTGPQLTDNNNSKSKLFCSTSDFSVGIVFLRFAHDATFMSIMVWTSVSMVLLHHRHCQRMQYIFTLNQDPRGQAE
TTATHITILMLVVTFGFYLLSLICIIFYTYFIYSHSLRHCNDILVSGFPTISPLLLTFRDPKGPSCSVFFNC'
```

##### 2. 去掉密码子后结果

```
: 'MSTHDTSLKTTEEVAFAQIILLQCQFGVGTANVFLFVYNFSPISTGSKQRPRQVILRHMAVANALTFLTIFPNNMMTFAPIIPQTDLKCKLEFFTRLVARSTNLCSTCVLSIHQFV
TLVPVNSGKILRASVTNMAASYSCYSCWFFSVLNNIYIPIKVTGPQLTDNNNSKSKLFCSTSDFSVGIVFLRFAHDATFMSIMVWTSVSMVLLHHRHCQRMQYIFTLNQDPRGQAE
TTATHITILMLVVTFGFYLLSLICIIFYTYFIYSHSLRHCNDILVSGFPTISPLLLTFRDPKGPSCSVFFNC'
```

##### 3. 凯撒加密创建加密字典

```
?: # test your function
encryption_key = 3
print(encode_dict(encryption_key))

{'a': 3, 'b': 4, 'c': 5, 'd': 6, 'e': 7, 'f': 8, 'g': 9, 'h': 10, 'i': 11, 'j': 12, 'k': 13, 'l': 14, 'm': 15, 'n': 16, 'o': 17, 'p': 18, 'q': 19, 'r': 20, 's': 21, 't': 22, 'u': 23, 'v': 24, 'w': 25, 'x': 26, 'y': 0, 'z': 1, ' ': 2}
```

Step 3:

## 4. 验证并解密

```
[65]: # Use caesar to encode message using encryption_key = 3, and save the result as encoded_message.
# Print encoded_message.
message = "hi my name is caesar"
encryption_key = 3
encoded_message = caesar(message, encryption_key)
print(encoded_message)
# the result should be "klcpacqdpchlvcfdhvdv"

klcpacqdpchlvcfdhvdv
```

Step 4:

```
[66]: # Use caesar to decode encoded_message using encryption_key = -3.
# Store your decoded message as decoded_message.
# Print decoded_message. Does this recover your original message?
# your code here
decoded_message = caesar(encoded_message, -3)
print(decoded_message)

hi my name is caesar
```

总结：

在本次实验中，主要利用了字符串，数组，文件 io 相关的知识，其中

`res=[table[x] if x in table else x for x in genlist]`

可以快速根据字典翻译列表，可以经常用到