

Algoritmos para Análise de Sequências Biológicas

Ficha 3

Objetivo

- Dogma central
- Tradução
- Reading frames
- Listas

Dogma Central

Dogma central

DNA => RNA => Aminoácidos

DNA => RNA

Transformar T em U

DNA => Aminoácidos

- Agrupar 3 a 3
- Usar a tabela de tradução para transformar o codão no aminoácido correspondente

Proteínas

- Começa por um codão **M**
- Acaba por um codão de **Stop**
- Não pode conter codões de **Stop**

Tabela de Conversão

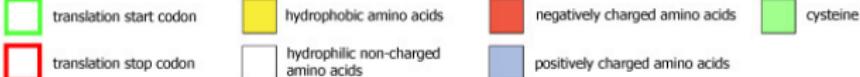
The Standard Genetic Code				
	U	C	A	G
U	UUU → Phe F UUC → Phe F UUA → Leu L UUG → Leu L	UCU → Ser S UCC → Ser S UCA → Ser S UCG → Ser S	UAU → Tyr Y UAC → Tyr Y UAA → Stop UAG → Stop	UGU → Cys C UGC → Cys C UGA → Stop UGG → Trp W
C	CUU → Leu L CUC → Leu L CUA → Leu L CUG → Leu L	CCU → Pro P CCC → Pro P CCA → Pro P CCG → Pro P	CAU → His H CAC → His H CAA → Gln Q CAG → Gln Q	CGU → Arg R CGC → Arg R CGA → Arg R CGG → Arg R
A	AUU → Ile I AUC → Ile I AUU → Ile I AUG → Met M	ACU → Thr T ACC → Thr T ACA → Thr T ACG → Thr T	AAU → Asn N AAC → Asn N AAA → Lys K AAG → Lys K	AGU → Ser S AGC → Ser S AGA → Arg R AGG → Arg R
G	GUU → Val V GUC → Val V GUA → Val V GUG → Val V	GCU → Ala A GCC → Ala A GCA → Ala A GCG → Ala A	GAU → Asp D GAC → Asp D GAA → Glu E GAG → Glu E	GGU → Gly G GGC → Gly G GGA → Gly G GGG → Gly G
 <ul style="list-style-type: none">translation start codon (green square)hydrophobic amino acids (yellow square)negatively charged amino acids (red square)cysteine (light green square)translation stop codon (red square)hydrophilic non-charged amino acids (white square)positively charged amino acids (blue square)				

Figure 1: Tabela de conversão de codão para aminoácido

Listas

Características

- Objeto com vários métodos

Métodos

`append` Permite adicionar um elemento no fim

`count` conta o nº de ocorrências de um elemento

`index` Devolve o 1º índice de um elemento ou -1

`pop` Remove um elmento por índice (por omissão o último)

Operadores

`+` Concatenação

`in` verifica se uma string está contida noutra

`[]` Índices sobre strings

Listas

Percorrer listas

```
for elt in lista:  
    fazer algo com o elt
```

Percorrer listas por índice

```
for idx in range(len(lista)):  
    fazer algo com lista[idx]
```

Percorrer listas por índice e valor

```
for idx, elt in enumerate(lista):  
    podemos usar o índice idx e o valor elt
```

Exemplos

```
>>> seq = "ACCTTGCA"
>>> l = []
>>> for x in seq:
...     l.append(x)
>>> l
['A', 'C', 'C', 'T', 'T', 'G', 'C', 'A']
>>> for i in range(0, len(seq), 3): print(seq[i : i + 3])
...
ACC
TTG
CA
```

Dicionários

Características

- Associa chaves a valores
- Cada chave só pode existir uma vez
- Os valores podem ser qualquer tipo de dado

Métodos

`get` Devolve o valor associado a uma chave ou o valor por omissão, caso exista

`keys` Devolve todas as chaves

`values` Devolve todos os valores

`items` Devolve todos os pares chave/valor

Exemplo

```
>>> d = {'rui' : 17, 'carla' : 19}
>>> for k, v in d.items(): print(k, v)
rui 17
carla 19
>>> [chave for chave in d]
['rui', 'carla', 'ana']
>>> d['ana'] = 14
>>> [(k, v) for k, v in d.items()]
[('rui', 17), ('carla', 19), ('ana', 14)]
```

Exemplo

```
>>> d['ana']
14
>>> d['dinis']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'dinis'
>>> d.get('dinis')
>>> d.get('dinis', "não tem")
'não tem'
```

Exercícios

Sugestões

- Os resultados devem ser sempre em maiúsculas
- Reutilize funções
- Crie funções auxiliares onde faça sentido
- Conceito avançado e poderoso: listas por compreensão

Exercícios

Exercícios

- ① Escreva a função `get_codons(dna)` que recebe uma sequência de DNA e devolve uma lista de codões
- ② Escreva a função `codon_to_amino(codons)` que recebe a lista de codões e devolve a sequência de aminoácidos (use `_` para o codão de **Stop**)
- ③ Escreva a função `get_prots(amino)` que recebe uma sequência de aminoácidos e devolve uma lista de possíveis proteínas
- ④ Escreva a função `get_orfs(dna)` que recebe uma sequência de DNA e devolve uma lista com as seis ORFs
- ⑤ Escreva a função `get_all_prots(dna)` que recebe uma sequência de DNA e devolve a lista com todas as proteínas

Exemplos de listas por compreensão

Carateres que não são bases de DNA

```
[x for x in dna.upper() if x not in 'ACGT']
```

Tradução numa linha

```
[codon_to_amino(cod) for cod in get_codons(dna)]
```

Todas as proteínas

```
[prot for orf in get_orfs(dna) for prot in get_prots(orf)]
```