

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

3D Molecular Designs		The Standard Genetic Code				Center for Biomolecular Modeling	
	U	C	A	G			
U	UUU → Phe F	UCU → Ser S	UAU → Tyr Y	UGU → Cys C	U		
	UUC → Phe F	UCC → Ser S	UAC → Tyr Y	UGC → Cys C	C		
	UUA → Leu L	UCA → Ser S	UAA → Stop	UGA → Stop	A		
	UUG → Leu L	UCG → Ser S	UAG → Stop	UGG → Trp W	G		
C	CUU → Leu L	CCU → Pro P	CAU → His H	CGU → Arg R	U		
	CUC → Leu L	CCC → Pro P	CAC → His H	CGC → Arg R	C		
	CUA → Leu L	CCA → Pro P	CAA → Gln Q	CGA → Arg R	A		
	CUG → Leu L	CCG → Pro P	CAG → Gln Q	CGG → Arg R	G		
A	AUU → Ile I	ACU → Thr T	AAU → Asn N	AGU → Ser S	U		
	AUC → Ile I	ACC → Thr T	AAC → Asn N	AGC → Ser S	C		
	AUA → Ile I	ACA → Thr T	AAA → Lys K	AGA → Arg R	A		
	AUG → Met M	ACG → Thr T	AAG → Lys K	AGG → Arg R	G		
G	GUU → Val V	GCU → Ala A	GAU → Asp D	GGU → Gly G	U		
	GUC → Val V	GCC → Ala A	GAC → Asp D	GGC → Gly G	C		
	GUA → Val V	GCA → Ala A	GAA → Glu E	GGA → Gly G	A		
	GUG → Val V	GCG → Ala A	GAG → Glu E	GGG → Gly G	G		

 translation start codon	 hydrophobic amino acids	 negatively charged amino acids	 cysteine
 translation stop codon	 hydrophilic non-charged amino acids	 positively charged amino acids	

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 elemento 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
```

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

- 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)]
```