MC202GH - Estrutura de Dados - Turmas G e H

Laboratório 2 - *Manipulação de strings em Bioinformática*

Docente: Marcelo da Silva Reis

Monitor PED: Matheus Abrantes Cerqueira

Monitores PAD: Andreas Cisi Ramos
Wallace Gustavo Santos Lima

1 de setembro de 2022

Data de entrega: 9/9/2022

Entrega no codePost[1]

Informações gerais

O presente laboratório tem como objetivo o aprofundamento da programação em C, com foco na manipulação de cadeias de caracteres (strings), o que inclui o tipo `char` e o uso da tabela ASCII. Para isso, são propostos alguns problemas clássicos de Bioinformática, nos quais é feito uso da representação de sequências genômicas e proteômicas como cadeias de caracteres. Serão fornecidos arquivos com protótipos a serem modificados e enviados na plataforma de avaliação. Cada questão tem seu próprio arquivo com função *main*, porém cada questão também tem suas funções a serem desenvolvidas dentro do arquivo *funcoes.c*, as quais não devem produzir saídas de texto.

Importante 1: neste laboratório será permitido o uso apenas das bibliotecas `stdio.h` e `math.h`. Em particular, **não** será permitido o uso da biblioteca **string.h**. Além disso, para compilar os códigos usando a biblioteca *funcoes.h* use o compilador `gcc` com as seguintes flags:

```
gcc -Wall -Werror -ansi -lm funcoes.c <program>.c -o <program> -I. funcoes.h
```

dentro do diretório que contenha esses arquivos, sendo que $< program >$ corresponde ao arquivo de cada questão.

Importante 2: Todas as instâncias que envolverem vetores de caracteres não excederão 1000 elementos. Por conta disso, recomenda-se que usem a diretiva `# define` para definir o tamanho máximo da sequência (`MAX_SEQ_SIZE` ou algo assim).

Importante 3: Deve-se considerar entradas maiúsculas e minúsculas nos experimentos (i.e. "TTC"equivale a "TtC"), sendo aconselhável criar uma função que converte as letras para o minúsculo. Porém não é necessário verificar se a entrada contém outros caracteres especiais diferentes dos especificados nesse roteiro.

[1] https://codepost.io/signup/join?code=ZW239C3IID

Questão 1 (1 ponto) - Contador de caractere em string

O objetivo desta questão é desenvolver um programa (*verifica_caractere.c*) que recebe um caractere c e um vetor de caracteres s e devolve um valor inteiro não-negativo i contendo o número de ocorrências de c em s.

Para esta questão, é necessário apenas implementar a captura de caractere e de string na função *main* e deve-se implementar a lógica da função:

int *analise_caractere(**char** c, **char** s[]);*

Questão 2 (3 pontos) - Tradução de sequências genômicas

Podemos representar tanto genes quanto proteínas como sequências de caracteres. No caso de genes, cada caractere representa uma base nitrogenada ou nucleotídeo (A, T, G ou C), enquanto que para proteínas cada um dos 20 aminoácidos é denotado por uma letra do alfabeto. Nas células de todos os seres vivos, genes são transcritos em moléculas de nucleotídeos conhecidas como RNA mensageiro (mRNA). Moléculas de mRNA, por sua vez, são traduzidas, em organelas chamadas ribossomos, em sequências protéicas. A tradução de nucleotídeos em aminoácidos é feita por trincas dos primeiros, de acordo com a tabela 1.

1st base	2nd base								3rd base
	T		C		A		G		
T	TTT	(Phe/F) Phenylalanine †	TCT	(Ser/S) Serine †	TAT	(Tyr/Y) Tyrosine †	TGT	(Cys/C) Cysteine †	T
	TTC		TCC		TAC		TGC		C
	TTA	(Leu/L) Leucine †	TCA		TAA	Stop (*Ochre*) *[note 2]	TGA	Stop (*Opal*) *[note 2]	A
	TTG →		TCG		TAG	Stop (*Amber*) *[note 2]	TGG	(Trp/W) Tryptophan †	G
C	CTT		CCT	(Pro/P) Proline †	CAT	(His/H) Histidine ‡	CGT	(Arg/R) Arginine ‡	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	(Gln/Q) Glutamine †	CGA		A
	CTG		CCG		CAG		CGG		G
A	ATT	(Ile/I) Isoleucine †	ACT	(Thr/T) Threonine †	AAT	(Asn/N) Asparagine †	AGT	(Ser/S) Serine †	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	(Lys/K) Lysine ‡	AGA	(Arg/R) Arginine ‡	A
	ATG →	(Met/M) Methionine †	ACG		AAG		AGG		G
G	GTT	(Val/V) Valine †	GCT	(Ala/A) Alanine †	GAT	(Asp/D) Aspartic acid ↓	GGT	(Gly/G) Glycine †	T
	GTC		GCC		GAC		GGC		C
	GTA		GCA		GAA	(Glu/E) Glutamic acid ↓	GGA		A
	GTG →		GCG		GAG		GGG		G

Tabela 1: Código genético, no qual são apresentadas a qual aminoácido corresponde cada tripla de nucleotídeos. Observe que TAA, TAG e TGA correspondem ao código de parada (stop), que é representado por um asterisco (*). Tabela extraída da Wikipedia.

Por exemplo, a seguinte sequência de nucleotídeos:

`GAATCCACCCGTGTTACCGTTCGTGCTTAA`

é traduzida para a seguinte sequência de aminoácidos:

`ESTRVTVRA*`

Escreva uma função (**void** *converte_nucleo*(**char n**[], **char a**[])) que recebe dois vetores de caracteres, sendo que o primeiro vetor (n) contém uma cadeia de nucleotídeos. A função deve traduzir essa cadeia em uma cadeia de aminoácidos, armazenando-a no segundo vetor (a).

Dica: escreva uma função auxilar, que recebe uma trinca de nucleotídeos e devolve o caractere do aminoácido correspondente.

Questão 3 (3 pontos) - Busca de motivos (motifs)

Motivos (*motifs*, em inglês) são pequenos padrões de sequências que se repetem em sequências maiores de nucleotídeos ou de aminoácidos. Em proteínas, motivos podem estar associados a função bioquímica das mesmas. Para fins de simplicidade, vamos supor aqui que um motivo é definido por uma única sequência de aminoácidos (em situações realísticas um motivo pode ter troca de aminoácidos em algumas posições da sequência).

Escreva uma função (**int** *freq_motivo*(**char** p[], **char m**[])) que recebe dois vetores de caracteres, p (sequência de nucleotídeos de um gene) e m (sequência de aminoácidos do motivo), e que devolve um inteiro não-negativo contando o número de ocorrências de m em na tradução de p para aminoácidos. O seu programa deve supor que a cadeia de caracteres em p tem tamanho igual ou maior do que a contida em m. Por exemplo, se p é igual a:

```
GAATCCACCCGTGTTACCGTTCGTGCTTTTTTTTTTTTTTTTTTTTTGAATCCACCCGTGTTACCGTTCGTGCTTAAA
AAAAAAAAAAAAAAAAAAAAAAAGAATCCACCCGTGTTACCGTTCGTGCT
```

e m é dado por:

```
ESTRVTVRA
```

Então a sua função deverá devolver 3.

Dica: para resolver este item generalize a função da Questão 1 e utilize a função de tradução implementada na Questão 2.

Questão 4 (3 pontos) - Cálculo de conteúdo de GC

Uma estatística muito importante no estudo de sequências genômicas é o cálculo da proporção de citosina (C) ou guanina (G) em uma dada sequência. Essa proporção também é conhecida como "conteúdo de GC", e serve para muitas aplicações, tais como desenho de primers para amplificação de DNA (utilizado, por exemplo, no teste de PCR para COVID-19). Embora essa estatística possa ser aplicada sobre todo o genoma, uma técnica importante é o cálculo do conteúdo de GC para uma determinada "janela" da sequência. Essa janela é então "deslizada" ao longo da sequência, gerando assim uma espécie de "média móvel" de conteúdo de GC.

Escreva uma função (**void** *conteudo_gc*(**char** n[], **int** j)) que recebe um vetor de caracteres contendo uma sequência de n nucleotídeos e um inteiro não-negativo j, $j \leq n$, e imprime na saída padrão, com duas casas decimais de precisão, o conteúdo de GC para uma janela de tamanho j que é deslocada de uma em uma posição, iniciando no elemento zero do vetor. Por exemplo, se o vetor de caracteres conter a sequência:

```
AATTGCGCAA
```

deverá ser impressa na tela a sequência de números reais (considerando $j = 4$):

```
0.00 0.25 0.50 0.75 1.00 0.75 0.50
```