Lista de Exercícios 2

Valor: 20 pontos
Questões: 15 (1 ponto cada) + 1 (5 pontos)

Descrição:

Algumas questões possuem uma descrição em itálico que servem como uma introdução do problema.

Bioinformatics Stronghold

1. (DNA) Faça um programa que leia um arquivo contendo uma *string* de DNA com o tamanho de até 1000 nucleotídeos. O programa deve retornar 4 inteiros separados por espaços sendo eles a quantidade de ocorrências de 'A', 'C', 'G', and 'T' respectivamente [1].

Exemplo entrada:
```
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC
```
Saída:
```
20 12 17 21
```

2. (RNA) Faça um programa que leia um arquivo contendo uma string de DNA. O programa deve retornar uma string que corresponda ao RNA transcrito do DNA informado, substituindo os “T”s por “U”s [2].

Exemplo entrada:
```
GATGGAACTTGACTACGTAAATT
```
Saída:
```
GAUGGAACUUGACUACGUAAAUU
```

3. (REVC) Faça um programa que leia um arquivo contendo uma string de DNA. O programa deve retornar uma string que corresponda ao DNA complementar (cDNA) do DNA informado [3].

Exemplo entrada:
```
AAAACCCGGT
```
Saída:
```
ACCGGGTTTT
```

4. (FIB) *A sequência de Fibonacci pode ser utilizada para estimar a população de coelhos a partir de uma quantidade de meses.* Considerando isso. faça um programa que leia um arquivo contendo dois inteiros (***n*** e ***k***) separados por espaço, onde ***n*** corresponde a quantidade de meses e ***k*** corresponde a quantidade de pares de filhotes de coelhos que cada casal (par) de coelhos adultos conseguem gerar por mês (ao invés de 1 par, como é normalmente na sequência de Fibonacci). Além disso, os coelhos filhotes demoram 1 mês para se tornarem adultos e capazes de se reproduzir. Sendo assim, o programa deve retornar o total de pares de coelhos depois de ***n*** meses, considerando inicialmente que existem 1 par de coelhos filhotes no primeiro mês [4].

Exemplo entrada:

```
5 3
```

Saída:

```
19
```

5. (GC) *O conteúdo GC de uma string de DNA é dado pela porcentagem de símbolos na string que são 'C' ou 'G'. Por exemplo, o conteúdo de GC de "AGCTATAG" é de 37,5%. Observe que o complemento reverso de qualquer string de DNA tem o mesmo conteúdo de GC.* Considerando isso, faça um programa que leia um arquivo FASTA contendo várias identificadores (que começam com o carácter “>”) e sequências de DNA. O programa deve retornar o identificador da sequência que possua o maior conteúdo GC seguido do valor do conteúdo GC [5].

Exemplo entrada:

```
>Rosalind_6404
CCTGCGGAAGATCGGCACTAGAATAGCCAGAACCGTTTCTCTGAGGCTTCCGGCCTTCCCTCCCACTAATAATTCTGAGG
>Rosalind_5959
CCATCGGTAGCGCATCCTTAGTCCAATTAAGTCCCTATCCAGGCGCTCCGCCGAAGGTCTATATCCATTTGTCAGCAGACACGC
>Rosalind_0808
CCACCCTCGTGGTATGGCTAGGCATTCAGGAACCGGAGAACGCTTCAGACCAGCCCGGACTGGGAACCTGCGGGCAGTAGGTGGAAT
```

Saída:

```
Rosalind_0808
60.919540
```

6. (HAMM) *A distância de Hamming (dH) entre duas cadeias com o mesmo comprimento é o número mínimo de substituições de símbolos necessárias para transformar uma cadeia na outra.* Sendo assim, faça um programa que leia um arquivo contendo 2 cadeias de DNA ***s*** e ***t*** de tamanho idêntico. O programa deve retornar a distância de Hamming entre ***s*** e ***t***, denotada por ***dH(s,t)***, onde ***dH(s,t)*** é o número de símbolos correspondentes que diferem em ***s*** e ***t*** [6].

Exemplo entrada:

```
GAGCCTACTAACGGGAT
CATCGTAATGACGGCCT
```

Saída:

```
7
```

7. (PROT) *Os 20 aminoácidos de ocorrência comum são abreviados usando 20 letras do alfabeto inglês (todas as letras, exceto B, J, O, U, X e Z). As cadeias de proteínas são construídas a partir desses 20 símbolos. Doravante, o termo string genético irá incorporar cadeias de proteínas junto com strings de DNA e strings de RNA.*
A tabela de codificação de RNA determina os detalhes relativos à codificação de codões específicos no alfabeto de aminoácidos. Sendo assim, faça um programa que leia um arquivo contendo uma string de RNA (mRNA). O programa deve retornar uma string que corresponda a sequência proteica codificada do mRNA informado [7].

Exemplo entrada:
AUGGCCAUGGCGCCCAGAACUGAGAUCAAUAGUACCCGUAUUAACGGGUGA
Saída:
MAMAPRTEINSTRING

8. (SUBS) Faça um programa que leia um arquivo contendo uma string de DNA ***s*** e outra ***p***. O programa deve retornar todas as posições (começando e 0) ao qual foi encontrado ***p*** em ***s***, separadas por espaços [8].

Exemplo entrada:
GATATATGCATATACTT
ATAT
Saída:
2 4 10

9. (SPLC) *Após de identificar os exons e os introns de uma string de RNA, precisamos apenas deletar os introns e concatenar os exons para formar uma nova string pronta para a tradução.* Considerando isso, faça um programa que leia um arquivo FASTA contendo no primeiro identificador uma cadeia de DNA ***s*** seguida de um coleção de substrings de ***s*** que atuam como introns. O programa deve retornar o uma sequência de aminoácidos resultante da transcrição e tradução dos exons de ***s*** [9].

Exemplo entrada:
>Rosalind_10
ATGGTCTACATAGCTGACAAACAGCACGTAGCAATCGGTCGAATCTCGAGAGGCATATGGTCACATGATCGGTCGAGC
GTGTTTCAAAGTTTGCGCCTAG
>Rosalind_12
ATCGGTCGAA
>Rosalind_15
ATCGGTCGAGCGTGT
Saída:
MVYIADKQHVASREAYGHMFKVCA

Bioinformatics Textbook Track

10. (BA1A) *Podemos dizer que um padrão **p** é o mais frequente **k**-mer (onde **k** é o tamanho de **p**) em um texto **t** se a quantidade de **p** em **t** for maior que todos os **k**-mers. Por exemplo, **"ACTAT"** é um 5-mer mais freqüente em "AA**ACTA**TAC**ACTA**TA**ACTA**TT", e **"ATA"** é um 3-mer mais freqüente de "CG**ATATA**TCC**ATA**G".* Considerando isso, faça um programa que leia um arquivo contendo uma string de DNA **s** seguida de um inteiro **k**. O programa deve retornar todos os **k**-mers mais frequentes em **s**, separados por espaço [10].

Exemplo entrada:

```
ACGTTGCATGTCGCATGATGCATGAGAGCT
4
```

Saída:

```
CATG GCAT
```

Classic Bioinformatics

11. Faça o programa que leia o resultado de alinhamento entre duas sequências com BLAST e retorne a identidade (em porcentagem). Dica: copie a tabela abaixo e cole em um arquivo chamado "resultado.txt"; leia o arquivo manualmente e identifique qual linha armazena o valor de identidade; depois crie um programa que leia e imprima na tela apenas esse valor.

```
Query= sp|P52407|E13B_HEVBR Glucan endo-1,3-beta-glucosidase, basic
vacuolar isoform OS=Hevea brasiliensis OX=3981 GN=HGN1 PE=1 SV=2

Length=374

Subject= sp|Q9SE50|BGL18_ARATH Beta-D-glucopyranosyl abscisate
beta-glucosidase OS=Arabidopsis thaliana OX=3702 GN=BGLU18 PE=1 SV=2

Length=528


 Score = 25.0 bits (53),  Expect = 0.005, Method: Compositional matrix adjust.
 Identities = 26/126 (21%), Positives = 51/126 (40%), Gaps = 18/126 (14%)

Query  233  FTSPSVVVWDGQR--GYK---NLFDATLDALYSALE------RASGGSLEVVVSESGWPS  281
            +T+ S+V WD +   GYK     F+  LD     L       + + G  EV+++E+G+
Sbjct  368  WTTDSLVDWDSKSVDGYKIGSKPFNGKLDVYSKGLRYLLKYIKDNYGDPEVIIAENGYGE  427

Query  282  AGA-------FAATFDNGRTYLSNLIQHVKGGTPKRPNRAIETYLFAMFDENKKQPEVEK  334
                      F     N + Y+   +  +     K        +++++ D  + Q   +
Sbjct  428  DLGEKHNDVNFGTQDHNRKYYIQRHLLSMHDAICKDKVNVTGYFVWSLMDNFEWQDGYKA  487

Query  335  HFGLFF  340
             FGL++
Sbjct  488  RFGLYY  493
```

12. Faça um programa que leia um arquivo FASTA contendo múltiplos identificadores e sequências. Para cada identificador e sequência o programa deve criar um arquivo de saída contendo o identificar e sua respectiva sequência. O nome dos arquivos de saída devem ser iguais aos seus respectivos identificadores, seguido de ".fasta".

Exemplo entrada:

>Rosalind_10
ATGGTCTACATAGCTGACAAACAGCACGTAGCAATCGGTCGAATCTCGAGAGGCATATGGTCACATGATCGGTCGAGC
GTGTTTCAAAGTTTGCGCCTAG
>Rosalind_12
ATCGGTCGAA
>Rosalind_15
ATCGGTCGAGCGTGT
Arquivos de saída:
Rosalind_10.fasta
Rosalind_12.fasta
Rosalind_15.fasta

13. Faça um programa que leia um arquivo contendo várias strings ***n***, onde ***n*** são os caminhos dos arquivos de entrada FASTA**.** O programa deve ler o conteúdo de cada arquivo de entrada e transformar em um único arquivo FASTA (nomeado como "all.fasta"), contendo todas os identificadores e sequências de cada arquivo de entrada lido.
Exemplo entrada:
Rosalind_10.fasta
Rosalind_12.fasta
Rosalind_15.fasta
Arquivo "all.fasta" de saída:
>Rosalind_10
ATGGTCTACATAGCTGACAAACAGCACGTAGCAATCGGTCGAATCTCGAGAGGCATATGGTCACATGATCGGTCGAGC
GTGTTTCAAAGTTTGCGCCTAG
>Rosalind_12
ATCGGTCGAA
>Rosalind_15
ATCGGTCGAGCGTGT

14. Faça um programa que leia um arquivo contendo várias linhas onde cada linha possui uma string ***a*** separada por espaço de uma string ***c***, onde ***a*** é um aminoácido em um determinado formato (**"nome","sigla","letra"**) e ***c*** é o formato a ser convertido de ***a***. O programa deve retornar todas as conversões de ***a*** para o formato ***c***. Segue Tabela de aminoácidos.

Exemplo entrada:
ALA nome
V sigla
Tirosina letra
Saída:
Alanina
Val
Y

15. *Estruturas de proteínas podem ser representadas por arquivos. Esse arquivos recebem o formato* ***PDB****, provindo de* ***P****rotein* ***D****ata* ***B****ank. Dentro de arquivos* ***PDB****s podemos encontrar muitas informações sobre uma proteína, como método experimental utilizado, atributos do experimento, posições espaciais do átomos dos aminoácidos e solvente presente, etc. Apesar do Protein Data Bank prover o formato* ***FASTA*** *(estrutura primária) de uma proteína em sua base de dados, nem sempre elas são idênticas a sequência encontrada no arquivo* ***PDB*** *e por isso para garantir essa exatidão é necessário realizar a conversão de um arquivo de*

*sequências de aminoácidos (**FASTA**) a partir do arquivo de estrutura (**PDB**) de uma proteína.* Sendo assim, faça um programa que leia um arquivo **PDB**. O programa deve converter o arquivo PDB para o formato FASTA, onde cada identificar do arquivo **FASTA** contenha a sequência de aminoácidos de cada cadeia polipeptídica do arquivo **PDB**.

Exemplo entrada:
1A1M.pdb
Saída:

```
>1A1M_A
GSHSMRYFYTAMSRPGRGEPRFIAVGYVDDTQFVRFDSDAASPRTEPRPPWIEQEGPEYWDRNTQIFKTNTQTYRENL
RIALRYYNQSEAGSHIIQRMYGCDLGPDGRLLRGHDQSAYDGKDYIALNEDLSSWTAADTAAQITQRKWEAARVAEQL
RAYLEGLCVEWLRRYLENGKETLQRADPPKTHVTHHPVSDHEATLRCWALGFYPAEITLTWQRDGEDQTQDTELVETR
PAGDRTFQKWAAVVVPSGEEQRYTCHVQHEGLPKPLTLRWEPHH
>1A1M_B
IQRTPKIQVYSRHPAENGKSNFLNCYVSGFHPSDIEVDLLKNGERIEKVEHSDLSFSKDWSFYLLYYTEFTPTEKDEY
ACRVNHVTLSQPKIVKWDRDM
>1A1M_C
TPYDINQML
```

16. Escreva um programa que leia um arquivo PDB e salve a sequência em formato FASTA. Teste com o PDB 2LZM (https://www.rcsb.org/structure/2lzm). (Valor: 5 pontos).