Instituto de Ciência e Tecnologia - UNIFESP
UC: introdução a bioinformatica
1º semestre de 2025
Professor: Thiago Martini Pereira

Trabalho Final de Algoritmos em Bioinformática

Objetivo Geral:

O trabalho da disciplina consiste em aplicar e implementar conceitos fundamentais de algoritmos em bioinformática para investigar a variabilidade genética de sequências virais. Os grupos deverão desenvolver as funções necessárias para as tarefas propostas, sem a utilização de bibliotecas de alto nível como Biopython para as funcionalidades centrais de análise de sequências e proteínas.

Estrutura da Entrega:

- Cada grupo deverá realizar uma apresentação oral sobre o projeto, discutindo a metodologia, os resultados e as conclusões.
- Ao final do projeto, o grupo deverá entregar ao docente:
    - O código Python desenvolvido em Colab (.ipynb). O código deve ser bem comentado, modular e demonstrar a implementação das funções solicitadas.
    - Um arquivo PDF contendo os slides da apresentação.

A apresentação e o código devem abordar as perguntas e análises descritas abaixo.

Atividade Final - Análise da Variabilidade Genética Viral:
Nesta atividade final, os grupos investigarão a variabilidade genética de sequências de nucleotídeos de diferentes isolados virais, que serão atribuídas a cada grupo. Utilize os alinhamentos e as análises subsequentes para identificar regiões conservadas e variáveis, e discuta o possível impacto dessas variações no contexto do vírus específico atribuído.

Conjuntos de Dados Disponíveis (Um será atribuído a cada grupo):

- (D1) Conjunto de Dados 1:Códigos de Identificação do GenBank: AY423387.1, AY423388.1,AB253429.1, AF004456, K03455.1
- (D2) Conjunto de Dados 2:Códigos de Identificação do GenBank: CY058444.1, CY033010.1, KF874987.1, MG009187.1, MK156321.1
- (D3) Conjunto de Dados 3:Códigos de Identificação do GenBank: DQ676839.1, GQ868512.1, JX144414.1, KX381014.1, EF440474.1

Instituto de Ciência e Tecnologia - UNIFESP
UC: introdução a bioinformatica
1º semestre de 2025
Professor: Thiago Martini Pereira

Instruções Específicas (com foco em implementação própria e contextualização biológica):

a) Download e Parse de Dados:
   a. Vá ao GenBank e realize o download do arquivo FASTA das sequências do conjunto de dados atribuído ao seu grupo.
   b. Desenvolva um código em Python para fazer a leitura das informações de cada arquivo FASTA. A função deve ser implementada pelo grupo, identificando o cabeçalho e a sequência de nucleotídeos. As informaçoes devem ser guardadas dentro de um dicionario

b) Descrição e Tamanho das Sequências:
   a. Para cada um dos registros solicitados, faça uma descrição das informações (organismo e origem).
   b. Desenvolva uma função em Python para determinar o tamanho de cada uma das sequências importadas.

c) Frequência de Nucleotídeos e Análise Comparativa:
   a. Implemente uma função em Python que calcule a frequência de cada nucleotídeo (A, T, C, G) em uma dada sequência.
   b. Com base nos resultados da sua função, faça um gráfico de barras com a frequência dos nucleotídeos de cada um dos registros.
   c. Coloque os nucleotídeos em ordem alfabética para facilitar a análise.
   d. Há diferenças nas frequências entre as sequências do vírus que seu grupo analisou? Qual poderia ser o motivo destas diferenças considerando o contexto de variabilidade viral (serotipos, origem geográfica, ano de isolamento, etc. conforme o vírus do seu conjunto de dados)?

d) Conteúdo GC e Temperatura de Melting (Tm):
   a. Implemente uma função em Python para calcular o conteúdo GC (porcentagem de G e C) de uma sequência de nucleotídeos.
   b. Utilizando o conteúdo GC, implemente uma função para calcular a temperatura de melting (Tm) de cada uma das sequências. Vocês deverão pesquisar e escolher uma fórmula apropriada para o cálculo da Tm (por exemplo, a fórmula de Wallace ou outra que considerem mais relevante para sequências curtas ou longas, dependendo da média dos tamanhos das sequências fornecidas). Justifique a escolha da fórmula.
   c. Discuta qual é a importância da temperatura de melting para a técnica de PCR, especialmente no diagnóstico ou caracterização de sequências do vírus que seu grupo está analisando?

e) Alinhamento Global (Needleman-Wunsch) e Identificação de Regiões:
   a. Implemente uma função que realize o algoritmo de alinhamento global (Needleman-Wunsch) em Python.

b. Utilize sua implementação para fazer o alinhamento global 2 a 2 dos primeiros 300 nucleotídeos das sequências. Esta limitação visa facilitar a implementação e execução dos algoritmos de programação dinâmica.
c. Regra de Pontuação: Utilize a seguinte regra de pontuação:
    i. Match (casamento): +1
    ii. Mismatch (descamento): -1
    iii. Gap Penalty (penalidade de gap): -2
d. Imprima e anote o score máximo de cada alinhamento, assim como sua similaridade (percentual de identidade).
e. Faça uma breve análise destes resultados, identificando se há regiões altamente conservadas ou variáveis nos primeiros 300 nucleotídeos e o que isso pode indicar sobre a função ou evolução do vírus que seu grupo está analisando.

f) Alinhamento Local (Smith-Waterman) e Destaque de Motivos:
    a. Implemente o algoritmo de alinhamento local (Smith-Waterman) em Python.
    b. Utilize sua implementação para fazer o alinhamento local 2 a 2 dos primeiros 300 nucleotídeos das sequências.
    c. Regra de Pontuação: Utilize a seguinte regra de pontuação:
        i. Match (casamento): +1
        ii. Mismatch (descamento): -1
        iii. Gap Penalty (penalidade de gap): -2
    d. Imprima e anote o score máximo de cada alinhamento, assim como sua similaridade (percentual de identidade) e as regiões alinhadas.
    e. Compare e discuta as diferenças entre os resultados do alinhamento global (Needleman-Wunsch) e do alinhamento local (Smith-Waterman) para estas sequências. Quais regiões de alta similaridade o Smith-Waterman destacou que talvez não fossem evidentes no alinhamento global? Em que situações cada tipo de alinhamento seria mais apropriado para investigar a variabilidade de sequências do vírus que seu grupo está analisando?

g) Síntese Proteica (Tradução) e Frequência de Aminoácidos:
    a. Implemente uma função em Python que realize a tradução de uma sequência de nucleotídeos (dada uma sequência de DNA ou RNA e uma tabela de códons). Para simplificar, realize a tradução a partir da primeira fase de leitura (iniciando no primeiro nucleotídeo), parando no primeiro códon de parada encontrado ou no final da sequência. Vocês podem pesquisar uma tabela de códons padrão para usar na implementação. Caso o codon não seja encontrado, identificar com X
    b. Construa um gráfico de barras com a frequência dos aminoácidos traduzidos para cada