

Documentação da Biblioteca Python Bioclass

Classe Principal

```
class Bioprof:
    """Bioprof - Objeto para tratamento de genomas"""

    def __init__(self):
```

Atributos da classe

```
self.__version__ = "0.0x"
self.id = "" #string que contém o nome(identificação da sequencia)
self.ids = [] #Lista de id's das sequências.
self.info = {} #Dicionario que armazena um comentário da sequência
self.composicao_total = [] #Totalização de nucleotídeos na sequência
self.sequencia = [] # Lista string com sequencia de DNA,RNA ou Proteína
self.alerta = True
self.codons = {"UUU": "F", "UUC": "F", ...
self.blosum_matrix = bl.BLOSUM(62) # Define a matriz blosum dentro
```

Métodos Públicos:

```
def leiaArquivoFasta(self,nome_arquivo):
    """ Lê todas as sequências de um arquivo Fasta """
```

```
def add_seq(self,n1,n2,n3): #n1=id, n2=comentário, n3=sequencia
    """Insere uma sequência através de código"""
```

```
def get_seqs(self):
    """Retorna a lista de sequencias registradas"""
```

```
def get_info(self,arg):
    """Retorna comentário da sequência"""
```

```
def get_tamanho_sequencia(self,arg):
    """Soma a quantidade nucleotídeos ou aminoácidos de uma sequência"""
```

```

def get_sequencia(self,id):
    """ Retorna a sequência armazenada, podendo ser um DNA, RNA ou Proteína """

def get_tipo_seq(self,id):
    """Identifica uma sequência se é de DNA, RNA ou PROTEINA"""

def get_percentual_GC(self,arg):
    """Retorna a percentual(%) de Guanina e Citosina na sequência."""

def insert_comment(self,id,info):
    """Armazena o comentário da sequência"""

def ver_info_seq(self,id):
    """Exibe dados (relatório) de uma sequência"""

def dH(self,id1,id2):
    """Distância de Hamming (dH) entre duas sequências de DNA ou RNA"""

def traduz_rna2proteina(self,id):
    """Traduz os códons de nucleotídeos em aminoácidos"""

def transc_dna2rna(self,id):
    """Transcrição de uma sequência de DNA em um mRNA"""

def get_kmers(self,id,k):
    """Retorna uma lista de sequencias k da janela deslizando da sequência

def get_assembly_kmers(self):
    """Assembly, monta um genoma a partir dos kmers """

    Desafio refazer! Aqui o método considera que mers estão em sequência

    #-----

#def matriz_d(self, filename):
    """Alinhamento de sequência """

```