PROG05 Tarea de evaluación 01. Realiza un programa en Java (100%)

Límite de entrega: miércoles, 18 de enero de 2023, 23:59
Ficheros requeridos: dna.txt, ecoli.txt (Descargar)
Número máximo de ficheros: 3
Tipo de trabajo: Individual

PROG05 TE01 REALIZA UN PROGRAMA EN JAVA (100%)

ENUNCIADO

Esta tarea se centra en el manejo de **arrays, ficheros y cadenas de caracteres** y necesitará 2 ficheros de texto, **dna.txt** y **ecoli.txt**, que se deberán guardar en la misma carpeta en la que guardamos el programa.

En los ficheros habrá información de muestras de ADN. Habrá 2 líneas por cada muestra. La primera línea contendrá el nombre de la muestra y la segunda línea la secuencia en sí. Las secuencias serán cadenas de caracteres que únicamente contendrán las letras 'A', 'C', 'G' y 'T' (pueden estar tanto en mayúsculas como en minúsculas) y el guion (-).

```
cure for cancer protein
ATGCCACTATGGTAG
captain picard hair growth protein
ATgCCAACATGgATGCCcGATAtGGATTgA
bogus protein
CCATt-AATgATCa-CAGTt
```

Cada letra representa un tipo de nucleótido y el guion (-) indica ADN "basura" o ADN no codificante.

Los nucleótidos son los elementos que forman el ADN y pueden ser de cuatro tipos diferentes: adenina (A), timina (T), citosina (C) y guanina (G). El ADN adopta una forma de doble hélice en la que una A siempre se enfrenta a una T y una C a una G.

En concreto el programa leerá estos ficheros que contienen secuencias de nucleótidos y generará información útil sobre ellos, simulando el trabajo de los biólogos computacionales.

Por cada secuencia de nucleótidos, se contará **cuántas veces aparece cada uno de los 4 tipos** (A, C, G y T) y se calculará el **porcentaje de masa** que corresponde a cada tipo de nucleótido **redondeado el resultado a un decimal.**

El programa también **mostrará los codones** en los que se organiza cada una de las secuencias. Es decir, **agrupará los nucleótidos de tres en tres** representándolos por sus iniciales. TAC, ATG o GGA serían ejemplos de codones.

Finalmente, indicará si la secuencia podría ser un gen que codifica una proteína o no.

Para ello, consideraremos que la secuencia es un gen si cumple las siguientes condiciones:

Comienza con un codón valido, es decir el primer codón es **ATG**
Finaliza con un codón válido, es decir el último codón será **TAA, TAG o TGA**
Contiene al menos 5 codones incluidos el de inicio y el de fin
Citosina (**C**) **y** Guanina (**G**) juntos **suponen al menos el 30% del total de la masa**.

El programa comenzará con una explicación de lo que va a hacer y pedirá 2 ficheros, un fichero de entrada con los datos de ADN que se quiere procesar y otro de salida donde se escribirán los resultados obtenidos. Se asume que el nombre del fichero de entrada introducido existe y cumple con el formato descrito.

```
Este programa genera información sobre
secuencias de nucleótidos de ADN contenidas en un fichero
También indicará si pueden codificar proteínas o no
Todos los resultados se guardarán en un fichero

Introduce el nombre del fichero: dna.txt
Introduce el nombre del fichero: output.txt
```

A continuación, por cada una de las secuencias mostrará:

La **descripción** y la **secuencia** que viene en el fichero
Cuántas veces aparece cada nucleótido: [A, C, G, T]
El porcentaje de masa de cada nucleótido respecto al **total**
Los codones de la secuencia
Si codifica una **proteína** o no

con el siguiente **formato**:

```
Descripción: cure for cancer protein
Nucleótidos: ATGCCACTATGGTAG
Contadores: [4, 3, 4, 4]
Masa (%): [27.3, 16.8, 30.6, 25.3] de 1978.8
Lista Codones: [ATG, CCA, CTA, TGG, TAG]
Es proteína: SI

Descripción: captain picard hair growth protein
Nucleótidos: ATGCCAACATGGATGCCCGATATGGATTGA
Contadores: [9, 6, 8, 7]
Masa (%): [30.7, 16.8, 30.5, 22.1] of 3967.5
Lista Codones: [ATG, CCA, ACA, TGG, ATG, CCC, GAT, ATG, GAT, TGA]
Es proteína: SI

Descripción: bogus protein
Nucleótidos: CCATT-AATGATCA-CAGTT
Contadores: [6, 4, 2, 6]
Masa (%): [32.3, 17.7, 12.1, 29.9] de 2508.1
Lista Codones: [CCA, TTA, ATG, ATC, ACA, GTT]
Es proteína: NO
```

Tened en cuenta que **la secuencia de nucleótidos** se muestra siempre en **mayúsculas** y que la **cantidad y la masa** de cada nucleótido se muestran en el siguiente **orden: A, C, G y T**. Fijaos también que un mismo codón puede aparecer más de una vez en la misma secuencia.

Para **calcular** los **porcentajes de masa**, se deberán tener en cuenta los siguientes valores de las masas unitarias:

```
Adenina (A): 135.128
Citosina (C): 111.103
Guanina (G): 151.128
Timina (T): 125.107
"Basura" (-): 100.000
```

Con estos datos calcularemos la **masa total** de una secuencia y los **porcentajes** de cada nucleótido.

Por ejemplo si la secuencia es: ATGG-AC

```
La masa total será: 135.128 + 125.107 + 151.128 + 151.128 + 100.000 + 135.128 + 111.103 = 908,722
Adenina (%): 2 x 135,128 * 100 / 908,722 = 29.7
Citosina (%): 111,103 x 100 / 908,722 = 12.2
Guanina (%): 2 x 151,128 x 100 / 908,722 = 33.3
Timina (%): 125.107 x 100 / 908,722 = 13.8%
"Basura" (%): 100.000 x 100 / 908,722 = 11.0
```

El programa utilizara al menos las siguientes **constantes**:

Número mínimo de codones de una proteína: 5
Porcentaje de masa de Citosina y Guanina juntas: 30
Número de tipos de nucleótidos: 4
Número de nucleótidos por codón: 3

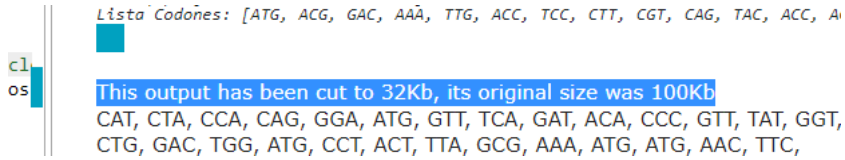
y seguirá funcionando correctamente si se **cambia el valor de las 2 primeras.**

Al menos se utilizarán **4 métodos con parámetros,** al menos uno de ellos tendrá que **devolver un array** y otro **recibirá un array** como parámetro. Recuerda que los arrays se pasan por referencia.

Se creará el **método escribirFichero**, al que se le llamará **una vez para escribir todos los resultados obtenidos** a partir de una secuencia de nucleótidos en el **fichero** y otra para mostrarlos por **consola**. El **resto de métodos** podrán realizar los cálculos que permitirán ir obteniendo todos los valores que necesitará el método anterior.

AVISO IMPORTANTE

Si al pasar el 2º test os sale el siguiente **error**:



Revisad la escritura de la secuencia por consola, **System.out.println** podría ser el origen del error. Utilizad **PrintStream** en su lugar.

RECURSOS

Para realizar esta tarea puedes utilizar jGRASP o el editor VPL de Moodle que además te dirá si la salida del programa coincide con la propuesta.

INDICACIONES DE DESARROLLO

Para procesar cada una de las secuencias de nucleótidos **se utilizarán al menos los siguientes arrays**. Pueden utilizarse más si se ve conveniente:

para llevar la **cuenta de cada tipo de nucleótido**
para los **porcentajes de masa**
para la **lista de codones**.

Realiza el programa por partes. Escribe el código poco a poco compilándolo y testeándolo frecuentemente.

Empieza por leer el fichero de entrada. Lee la descripción y la secuencia y muéstralas por pantalla.
Luego recorre la secuencia de nucleótidos y cuenta el número de As, Cs, Gs, y Ts. Utiliza un array para llevar la cuenta. Te puede interesar crear un método que convierta cada una de las letras a un índice del 0 al 3.
Calcula la masa total y después calcula el array con los porcentajes de masa de cada nucleótido. Si quieres en este caso, puedes almacenar los valores redondeados.
Ten en cuenta que el ADN "basura" (-) contribuye a la masa total pero en otras partes del programa te puede interesar eliminarla.
A continuación, se separará la secuencia de nucleótidos en grupos de 3 (codones) y se guardarán en un array. Se asumirá que cada secuencia de nucleótidos sin guiones será múltiplo de 3. Es decir, se dividirá en codones completos.
Se analizará dicho array y el de porcentajes de masa para comprobar si cumple con las condiciones de una proteína o no.
Consigue que el programa primero muestre todos los resultados correctamente por consola antes de guardarlos en el fichero. Se asumirá que si el fichero de salida existe se sobrescribirá.

Utiliza métodos, parámetros y returns para estructurar el programa y evitar redundancias.

Para conseguir la máxima nota, se utilizarán al menos 4 métodos con parámetros. Al menos uno devolverá un array y otro recibirá un array como parámetro.
También se creará el método escribirFichero, al que se le llamará 2 veces: una para escribir todos los resultados en el fichero y otra para mostrarlos por consola. El resto de métodos podrán realizar los cálculos que permitirán ir obteniendo todos los valores que necesitará el método anterior.
Esta vez se puede utilizar println en el main, siempre que el main esté bien estructurado y sea un resumen conciso del programa.
También es correcto usar algún bucle en el main, siempre que el código dentro de él sea corto y legible.

Presta especial atención a las redundancias. Si tienes código que se repite, introdúcelo dentro de un método o un bucle para escribirlo una sola vez. Si tienes un método largo o incoherente, divídelo en partes más pequeñas.

Para resolver el programa necesitarás usar diferentes **métodos de la clase String**

Lee y escribe ficheros mediante objetos de la clase Scanner y PrintStream, pasándoles parámetros File.

Se asume que los nombres de fichero que se van a introducir son siempre válidos.
Se asume que todos los ficheros son válidos y cumplen con el formato descrito. Siempre tendremos una descripción y una secuencia.
Se utiliza la sentencia try-catch para evitar que el programa falle por una excepción mientras se lee o escribe en los ficheros

Guía de estilo (Referencia):

Da nombres significativos a métodos, variables y parámetros siguiendo además las reglas de Java.
Utiliza correctamente la indentación y los espacios en blanco.
Incluye cabeceras explicativas al principio del programa y al principio de cada uno de los métodos, así como junto a cualquier línea de código que lo pueda necesitar.
En el caso de los métodos, como comentario, además de lo que hace indica los parámetros que utiliza y el valor que devuelve si es que existe.

Penaliza: Utilizar recursos que no se han trabajado en las UDs anteriores o actual. No cumplir con las correcciones e indicaciones descritas en las Tares evaluativas anteriores (feedback de la calificación).

INDICACIONES DE ENTREGA

El programa se escribirá directamente en el IDE proporcionado o se copiará en él una vez finalizado.

En la cabecera, además de la información que describe el programa, se añadirá el enlace a la evaluación realizada, un enlace a Google Drive o a Youtube según el formato elegido

CRITERIOS DE CALIFICACIÓN. TOTAL 10 PUNTOS.

La tarea se evaluará siguiendo la siguiente rúbrica:

Funcionalidad (70%).

Comentarios (15%).

Legibilidad (15%).

Con esta rúbrica se calculará la nota de vuestro programa y se comparará con la que habéis indicado en la autoevaluación.

Si la diferencia es menor de un punto, se calculará la nota media entre las dos.
Si la diferencia es mayor, se utilizará directamente la calificación de la profesora.

En los casos en los que falte la autoevaluación, se utilizará la calificación de la profesora con una penalización de un punto.

Ficheros requeridos

dna.txt

```
cure for cancer protein
ATGCCACTATGGTAG
captain picard hair growth protein
ATgCCAACATGgATGCCcGATAtGGATTgA
bogus protein
CCATt-AATgATCa-CAGTt
michael jordan mad hops protein
ATgAG-ATC-CgtgatGTGgg-aT-CCTa-CT-CATTaa
paris hilton phony protein
AtgC-CaacaTGGATGCCCTAAG-ATAtgGATTagtgA
jimi hendrix guitar talent protein
atgataattagttttaatatcaga-ctgtaa
admiral grace murray hopper protein
ATGC-AATT--GC-----TCGA--------TTAG
tyler durden's brain protein
ATGATAcctatgagtaaTGTGGACCatatccaaACTATAGGCATtgtcggACCAACGATcgattggtTATACTGA
mini me growth hormone
AtGgGaCGCTgA
Nyan Cat protein
CAT-CAT-CAT-CAT-CAT-CAT-CAT-CAT-CAT-CAT
```

ecoli.txt

```
thr operon leader peptide
ATGAAACGCATTAGCaCCAcCATtACCACCaCCATCaCcATTACCACAGGTAACGGTGCGGGCTGA
aspartokinase I/homoserine dehydrogenase I
ATGCGAGtGTTGAAGTTcgGCGGTaCATCAgTGGCAAATGCAGAACGTtTTCTGCGGgTTGCCGATAttCTGGAAAGcAATGCCAGGCAGGGGCAGgTGGcCACCGTCCTCtCTGcCCCCGCCAAAATCACCAACCATCtGGTaGCGATGA
homoserine kinase
ATgGTTAAAgTTTAtGCCCCGGCtTCCAGTGCCaATATGaGcGTCGgGTTTGATGTGCTCGGGgCGGCGGTGACACCTGTTGATGGTGCATTGCTCGgAGaTGTagTcaCGGTTGAGGCGGCAGAGACaTTCAgTCTCAACAACCTCGGAC
threonine synthase
ATGAAACTCtacaATCTGAAAGATCACAATGAGCAGgTCaGCTTTGCGCAAGCCGTAACCCAGgGgTTAGGCAAAAATCAGGGgCtGTtTTTTCcgCACgaCCTGCCGGaaTTCAGCcTgACTGAAaTTGATGAGATgCTGAAGCtGGATT
hypothetical protein
AtGCAGCCcGGCTtTTTTTATGAAGAAAATaTGGAGaAaAACGACagGGAAAAAGGAGAAATTCtCAATAAATGCGGtAACTTAGAgATTaGGATTGCGGAGAATaACAACTGCcGTTCTCaTCGCGTAATCTCCGGATATCGACCCaTAA
Non-protein region
aAAAACTgCTGGAAACAATGAAAGAcGTACCGGACGACCAAcGTCAGgCGC
transaldolase B
ATGACGGACAAATTGaCCTCcCTTCGTCAGTACACCACCGTAgTGGCCGACACTGGGGACATCGCGGCAATGAAGcTGTaTCAACcGCAGGATGCCACAACCAAcCCTtCTCTCATTCTTAACGCAGCGCAGATTCcGGAATACCGTAAgT
molybdopterin biosynthesis mog protein
ATGAATACTTTACGTATTGGCTTaGTtTcCaTCTCTGATCGCGCATCCAGCGGCGTTTAtCAGgaTAAAgGCATCCCTGCGCTGGAagAATGGCTGACAtcGGCGCTAACCACGcCGTTTGAaCTGGAAAcCCgCTTaATCCCCGATGAGC
chaperone protein DnaK
aTGGGTAAAATAaTTGGTATCGACcTGGGTACtACCAaCTCTTGTGTagCGaTTAtGGATGGCACCACTCCtCGTGtACTGgAGAACGcCGAAGGCGATCGCACCAcGcCTTcTATCATTgCCTATACCCAGGAtGGTGAAACTCTGGTTg
chaperone protein DnaJ
GTGCatTCatCTAGGGGcAATTTAAAAAAGATGGCTAAGCAAGATTaTTACGAGaTTTTAGGCGTTTCCAAAaCAGCGGAAGAGCGtGAaaTCAAAAaGGCCTACAAACGCCTGGCCATGAAaTACCaCCCGGaCcGTAACCAGGgTGACA
hypothetical protein
TTGCTCTTaCTCGGATTCgTAAGCCGTGAAAACAGCAaCCTCCGtCTGGCCAGTTCGGATGTGAACCTCACAGAGgTCTTTTCTCGTTACCAgCGCCGCCACTACGGCGGTgATACAGATGACGATCAGgGcgACaAtcAtCgCcTTATGC
hypothetical protein
aTGTCTGCCAAaaGACGACTTCTTATTGCGtGTACCTTGAtAaCAGCTATcTATCAtTTTCCTGcaTATTCTTCATTAgAATATAAAGGAtCCTTTGGTTCAATaAATGCGGGTTAtGCAGACTGGAATAGTGGaTTTgTAAaCACTCACC
putative secreted sulfatase
ATGCAGAAAACGTTAATGGCCAGTTTGATCGGCCTTGCAGTTTGCACAGGGAAtGCTTTTAGtCCTGCCTTAGCCGCAGAGGCTaAACAACcTAATTTAGTCATtaTTATGGCGGaTGATtTAGGTtaTGGCGAtTAGcAaCaTATGGTC
putative cytoplasmic protein
ATGTTTACcAacGTAAATGTTGATTGtTgCAAAACACCAGGAtGTAAaaACCTGGGGTTGCTGAATAGCCAGGATTATGTCGCAcAGgGTaAaAATATTTtATGCCGTGAATGTgGTTaCTTGTtTCCAGtGATATCTGAACAGTCGCTTA
sodium/proton antiporter 1
GTGAAACATCTGcATCGATTCTTTAGCaGTGATGCCTCGGGAGgCATTATTCTCATTATTGCCGCTGTATTAGCGATGATTATGGCCAACAGCGGTgcAaCCAGTGGATGGTATCACGACTTTCTTGAGACGCcGGTTCAGcTcCGGGTTC
transcriptional activator protein NhaR
ATGAGCATGTCTCATaTCAATTACAACCACTtGTATTACTTCTGGCaTGTCTAcAAAgAaGGTTCTGtGGTTGGCgCAGCGGAGGCGCTTTATTTAACAcCAcAAACCATTACCGGGCaGATCCGGGCGCTGGAaGAGCGCCTGCAAGGGA
riboflavin kinase
ATGAAGCTGATACGCGgCAtACATAATCTCAGCCAGGCCCCGCAAGAAGGGTGTGTGCTGACTATTGGTaATTTCGACGGCGTGCATCGCggTCATCGCGCGCTGTTACAGGGCtTGCAGGAAGAAGGGCGCAAGCGCAACtTACCGGTGA
Isoleucyl-tRNA synthetase
ATGAGTGACTATAAATCaACCCTgAATTTGCCgGAAACAGgGTtCCCGATgCGTGGCGATCTCGcCAAGCGCGAAcCGGGaATGCTGGCGCGTTGGACTGATGATGATCTgTaCGGCATCATCCGTGCGGCTaAAAAAGGCAaAaAAACCT
Non-protein region
GCTTGCGCCAACGcCATTTCATCGCCATCCCGCCgAgcATACAGGCCTCgAaGAACCAaTGGTGTTGGTGcCAACGGCCtGAccATTTTTcGGTGCAGGCGCATGCCACAGATCGGCAACCATGTTTACGCAACGCAGATCGATTGCTGc
FKBP-type 16 kDa peptidyl-prolyl cis-trans isomerase
ATGTCTGAATCTGTACAGaGCAaTAgCGCCGTCCTGGTGCACTTCACGCTAAAACTCGACGAtGGCaCCAcCGCTGAGTCTACCCGCAaCAaCGGTAaACCGGCGCTGTTCCGCcTGgGTgATGCTTCTCTTTCTgAaGgGCTGGAGCAAC
```

VPL

Navega por la unidad

◄ PROG05 Tarea de aprendizaje 03. Arrays en métodos

Ir a...

UD05 Encuesta valoración ►

Contacta con nosotros:

Dirección: Paseo de Ubarburu 39, Edificio EnerTIC of. 206 · Donostia San Sebastián
Telefono : 945 567 953
E-mail: info@birt.eus
Twitter: @Birt_LH