Escuela de Computación
Bachillerato en Ingeniería en Computación

TEC | Tecnológico de Costa Rica

Proyecto #3

Fragmentos

IC-8022 Introducción a la Biología Molecular Computacional

Tecnológico de Costa Rica, Sede Central Cartago

Escuela de Computación, Ingeniería en Computación

I Semestre 2017

Prof. Ing. Esteban Arias Méndez

Un programa que genera hileras aleatorias de ADN con ciertas características controladas y un programa que fragmentará un archivo de texto en forma aleatoria, aunque también con ciertos parámetros definidos por el usuario.

DESARROLLO

Se trabajará sólo sobre el Sistema Operativo Linux, usando solamente lenguajes C o Python, plataforma web o móvil. Se recomienda trabajar su código en módulos, funciones o procedimientos, incluso en varios archivos para facilitar el trabajo. Debe hacer uso del git de la Escuela de Computación para llevar todo el desarrollo de su proyecto: git.ec.tec.ac.cr o uno alterno de fácil acceso.

Este trabajo es para ser desarrollado por parejas de 2 personas máximo.

**** Para todos los casos parametrizables, se debe mostrar al usuario el cálculo de cada operación, los valores de probabilidad obtenidos para cada caso, así como los datos originales a usar y cada uno de los cambios que se aplicarán y en qué lugares de la hilera. ****

Generador de hileras de ADN

Este programa genera una hilera aleatoria de ADN, pero con ciertas características controlables. Primero, se tiene una **hilera base** de la longitud indicada por el usuario. Después, esta hilera podrá ser enriquecida con ciertas características que el usuario establezca. Constantemente la hilera que se construye será desplegada al usuario para que éste pueda "navegar" en la misma. En cualquier momento, el usuario podrá grabar esta hilera en un archivo de texto con el nombre que desee.

Hilera Base

Esta es la hilera inicial del programa. Tiene la longitud que el usuario indique. Hay 4 modalidades posibles:

- **Ingresada por el usuario.** El usuario podrá teclear la hilera base o dar el nombre de un archivo que la contenga

- **Totalmente aleatoria.** Generada por el mismo programa en forma aleatoria. Los 4 símblos A, C, T, G tienen la misma probabilidad (0.25) de aparecer.
- **Bases desiguales.** Se permite que el usuario altere las probabilidades de aparición de cada base particular, por supuesto, siempre a suma total debe ser 1.0.
- **Probabilidades condionales** (*opcional, extra!*). El usuario podrá especificar la probabilidad para cada símbolo, dado que el símbolo anterior sea alguno particular.

Repeat after me …

La hilera de ADN generada podría presentar estructuras repetidas que el usuario puede inducir, o inclusive forzar. Un **repeat** es una subhilera de ADN de cierta longitud que aparece múltiples veces a lo largo de la hilera. Es posible que este *repeat* no esté perfecto todas las veces, sino que tenga ciertas alteraciones (inserciones, borrados, cambios de base, cambios de orientación, complemento, etc.). Además, una hilera podría presentar varios *repeats* diferentes. A continuación, se indican los parámetros que controlan a los *repeats*, nótese que la mayoría de estos parámetros son ortogonales entre ellos.

- **Fuente del repeat**: el usuario podría indicar el repeat que desea forzar o dejar que el programa seleccione aleatoriamente cualquier posición de la hilera actual. En el primer caso, el usuario podrá teclear el repeat o dar el nombre de un archivo de texto que lo contenga. En el segundo caso, el usuario indicará la longitud del *repeat* (la cual debe ser menor que la longitud actual de la hilera que se procesa) y el programa lo seleccionará aleatoriamente.
- **Cantidad de repeats**: el usuario podrá indicar exactamente cuántas veces quiere que el *repeat* sea insertado en la hilera base, o establecer que el programa debe insertar el *repeat* tantas veces como pueda en la longitud de la hilera actual.
- **Distancia entre repeats**: este parámetro establece cada cuanto aparecerá el *repeat*. Hay dos modalidades: distancia fija o distancia aleatoria. En el primer caso, el usuario dará la posición de la prime aparición del *repeat* y una distancia en bases que separa un *repeat* del siguiente. Cada *repeat* se **insertará** en la hilera actual en la posición que está exactamente a la distancia indicada del *repeat* previo. En el segundo caso, el usuario dará la distancia promedio en bases entre un

repeat y el siguiente. Siguiendo una distribución exponencial el programa generará distancias aleatorias.

- **Errores**: el usuario indicará cuales posibles errores de base podrían aparecer en el *repeat* (cambio de base, borrado, inserciones), y para cada uno de ellos establece la distancia promedio entre errores (distribución exponencial). Si no se indica ninguno de estos errores significa que el *repeat* no debe ser alterado antes de ser insertado.
- **Orientación inversa**: con cierta probabilidad que el usuario establece el *repeat* podría invertirse y complementarse antes de ser insertado. Si esta probabilidad es 0, el *repeat* nunca se invierte antes de insertarse.

Daba arroz a la zorra el Abad

La hilera de ADN podría mostrar estructuras palíndromas donde una región y su complemento inverso aparecen en el ADN separados por cierta distancia. El comportamiento general y las opciones deben ser similares a las dadas para los *repeats* en la sección anterior, pero con la diferencia que las inserciones son ahora en **pares**: la subhilera a ser insertada y a cierta distancia su complemento inverso. Igual al caso anterior el usuario podrá controlar la cantidad y distancia entre palíndromos y los posibles errores que éstos muestren. Hay que distinguir la distancia entre una hilera y su complemento inverso, y la distancia entre cada palíndromo – los cuales podrían estar inclusive anidados.

Archivo Descriptivo

Cuando el usuario solicite que la hilera actual sea grabada, se generarán en realidad dos archivos: uno con la hilera propiamente y otro archivo con información descriptiva de la misma. Ambos archivos podrían tener el mismo nombre con extensiones diferentes. Cada pareja puede definir el formato del archivo descriptivo, pero en todo caso deben documentarlo y buscar la mayor simplicidad posible. Por ejemplo, podría ser un archivo XML. Entre la información a guardar está : la hilera original, longitud original y actual de la hilera, características de la hilera base, si es dada o generada aleatoriamente, características de *repeats* y palíndromos, datos usados en cada caso, etc. Dicha información podría estarse generando cada vez que se hacen cambios en la hilera original. Este archivo debe planearse para permitir la "reejecución" del mismo para generar nuevamente una hilera con características equivalentes.

Modo Batch (opcional, extra!)

El programa generador ofrecerá un modo batch donde se generarán en forma no interactiva tantas hileras de ADN como el usuario indique, siguiendo los parámetros que se le den. Estos parámetros se dan en forma interactiva o cargándolos de un archivo descriptivo creado en una sesión previa. Todos estos archivos tendrán un nombre genérico indicado por el usuario como un parámetro adicional, seguido de un número consecutivo.

Generador de Fragmentos (shotgun)

Este programa interactivo tomará un archivo (que podría ser de ADN o un texto arbitrario) y generará fragmentos aleatorios siguiendo las especificaciones del usuario. Para el caso del ADN, existe también la posibilidad de introducir algunos errores en los fragmentos. Todos los fragmentos se grabarán en un archivo de texto con un formato a definir por cada grupo de trabajo.

Opciones generales

El usuario indicará cuantos fragmentos desea obtener (en realidad este es un número mínimo de fragmentos). Esto lo podrá hacer pidiendo explícitamente la cantidad de fragmentos o indirectamente estableciendo una cobertura promedio deseada. También se indicará la longitud promedio de los fragmentos y la desviación estándar de los mismos. Para este proyecto basta con seguir una distribución uniforme de las longitudes de los fragmentos, pero *opcionalmente* (extra!) se podrían ofrecer otras distribuciones (distribución normal, por ejemplo).

El usuario tendrá la opción de solicitar que los fragmentos **cubran totalmente** a la hilera base, *i.e.* que se garantice que para toda posición *i* de la hilera original hay al menos un fragmento que la contiene. Si no se pide esta opción, el programa no tiene la obligación de garantizar este requisito.

Opciones adicionales

El usuario podrá establecer algunos parámetros adicionales al programa, que principalmente introducen errores en los fragmetos.

- **Errores**: el usuario indicará cuales posibles errores de base podrían aparecer en cada fragmento (cambio de base o letra, borrado, inserciones) y para cada uno de ellos establece la distancia promedio entre errores (distribución exponencial). Si no se indica ninguno de estos errores o se indican en 0, significa que los fragmentos no serán alterados.
- **Quimeras**: este es un porcentaje que indica cuantos de los fragmentos generados serán quimeras (concatenación arbitraria de 2 o más fragmentos, los cuales a su vez podrán tener o no errores u orientaciones arbitrarias). El usuario indicará este porcentaje (de ser 0 no habrá quimeras) y la cantidad máxima de fragmentos a ser concatenados.
- **Orientación inversa**: con cierta probabilidad que el usuario establece el fragmento podría invertirse y complementarse. Si esta probabilidad es 0, la colección completa de fragmentos mostrará siempre la misma orientación.

Archivo Descriptivo

Cuando el usuario solicite que los fragmentos sean generados, se crearán dos archivos: uno con los fragmentos propiamente y otro archivo con información descriptiva de la colección. Ambos archivos podrían tener el mismo nombre con extensiones diferentes. Cada pareja puede definir el formato del archivo descriptivo, pero en todo caso deben documentarlo y buscar la mayor simplicidad posible. Por ejemplo, podría ser un archivo XML. Entre la información a guardar está : la cantidad de fragmentos, longitud promedio, desviación estándar, cobertura promedio, cobertura total, tipos de errores con sus probabilidades, presencia de quimeras y sus valores dados, orientación, etc. Este archivo debe planearse para permitir la "reejecución" del mismo para generar una nueva colección de fragmentos con características equivalentes.

Modo Batch (opcional, extra!)

El programa generador ofrecerá un modo batch donde se generarán en forma no interactiva tantas colecciones de fragmentos viniendo del mismo

archivo base como el usuario indique, siguiendo los parámetros que se le den. Estos parámetros se dan en forma interactiva o cargándolos de un archivo descriptivo creado en una sesión previa. Todos estos archivos tendrán un nombre genérico indicado por el usuario como un parámetro adicional, seguido de un número consecutivo.

Ensamblaje de Fragmentos (extra)

Utilizando alguno de los algoritmos de ensamblaje, tomar algún archivo de fragmentos, para procesarlo y reconstruir la hilera original siguiendo el algoritmo. Hacer una comparación de similitud entre la hilera original y la hilera reconstruida para su evaluación.

Brindar múltiples corridas (experimentos) del programa para valorar su efectividad, proveyendo cambios en los parámetros usados para los fragmentos, para valorar su valor de convergencia de acuerdo a los cambios introducidos en los fragmentos.

EVALUACIÓN

1. Rúbrica de evaluación:

    - El proyecto se calificará con los siguientes criterios:

        i. 80% - programas solicitados, interactividad, uso de parámetros, funciones de probabilidad usadas, generación de archivos descriptivos para su reejecución.
        ii. 20% - Documentación completa del trabajo.
        iii. 20% - extras, 10% ensamblaje, 10% modos batch y otros

2. El proyecto debe resolverse, implementándolo de la mejor manera.

3. De forma global, se evaluará la presentación del trabajo según los parámetros solicitados, estrategias empleadas y la calidad, la entrega a tiempo del trabajo y la documentación completa correspondiente.

4. Sobre la documentación y presentación:

    a. 2pts - El subject del correo a ser enviado debe ser:
    [BMC] – Proyecto # 3 – Sus Nombres Completos
    b. 2pts - El correo debe contener de forma separada:
        i. los archivos de texto de los códigos fuentes que permiten la solución y funcionalidad del mismo.
        ii. un archivo PDF con la documentación completa

    No envíe archivos ejecutables o binarios.

    c. La documentación en PDF con el nombre de archivo igual al subject del correo enviado. Esta documentación debe tener un apartado, que indique los pasos a seguir, para poder ejecutar el código (librerías a instalar y otros) en caso de usar herramientas adicionales a las brindadas.
        i. 5pts – La documentación debe incluir una portada con los datos completos: TEC, carrera, sede, curso y código, profesor, periodo, fecha de entrega, número de proyecto, título del proyecto, nombres completos con número de carnet de la pareja de trabajo y un abstract en inglés en la misma portada.

ii. 5pts – La introducción del documento es una descripción breve del trabajo realizado y herramientas usadas. Como mini-marco teórico incluya las referencias a los algoritmos implementados y las herramientas empleadas, así como otras fuentes de consulta utilizadas.
iii. 10pts – Como desarrollo debe explicar, los procedimientos, rutinas, la lógica que utilizo para resolver el problema. indicar los ejemplos de código que ha usado como guía para el desarrollo de los mismos usando las referencias bibliográficas correspondientes. Explicar el uso y funcionamiento.
iv. 20pts – Análisis de resultados, explicando el trabajo implementado, la forma de realización, funcionamiento, general, ejemplos documentados, problemas presentados, estructuras de datos empleadas, algoritmos usados, etc.
v. 10pts – Una sección de conclusiones y/o observaciones sobre el proyecto.
vi. 10pts – En una sección de Apéndices incluya el código fuente documentado del proyecto y su explicación. En caso que haya hecho cambios al código de terceros usado, indicar los cambios hechos. Explique la estructura del código empleada en su proyecto, módulos, etc.

5. La tarea puede realizarse de forma individual o en parejas de 2 máximo.

6. Se debe entregar en digital a más tardar el Vierens 16 de Junio, antes de la media noche. Debe hacerlo de forma simultánea a los correos siguientes y copiarse usted mismo y su compañero de trabajo. Cada día de atraso serán 15pts menos de la nota de la tarea:

    a. earias@ic-itcr.ac.cr
    b. kecastro@ic-itcr.ac.cr

7. Cualquier consulta puede hacerla al foro, o personalmente en clase o al correo del profesor con copia al asistente a los correos anteriores.

8. Durante la revisión del proyecto deben estar presentes ambos miembros de la pareja de trabajo, la no presentación les restará 10pts a cada uno de los ausentes.