Escuela de Computación
Bachillerato en Ingeniería en Computación

TEC | Tecnológico de Costa Rica

Proyecto #2

Ensamblaje de Fragmentos

IC-8050 Introducción a la Biología Molecular Computacional

Tecnológico de Costa Rica, Sede Central Cartago

Escuela de Computación, Ingeniería en Computación

II 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, para luego reconstruir el texto original.

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

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

Parámetros

Su programa debe poder parametrizarse.

Debe recibir los valores de probabilidad para cada uno de los 5 posibles errores: sustitución, inserción, deleción, quimeras, inversión. En caso de ser 0 significa que ese error no sucede.

Además, debe parametrizar el valor de % de cobertura deseado y el rango de valores de traslape mínimo a máximo, donde el valor dado entre estos límites puede ser aleatorio.

Generador de Fragmentos (shotgun)

Su programa debe tomar un archivo (que podría ser de ADN o un texto arbitrario) y generará fragmentos aleatorios siguiendo las especificaciones del usuario.

Existe también la posibilidad de introducir cualquiera de los errores en los fragmentos de acuerdo a los parámetros. El usuario seleccionará los tipos de errores que desea y los porcentajes de probabilidad en cada caso usando como base los parámetros iniciales dados. 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). Estos valores deben contar con valores por default dentro de sus parámetros al iniciar el programa.

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 fragmentos.

- **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 (en caso de ADN). 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

Su programa debe cargar un archivo de fragmentos, ya sea un archivo generado por ustedes o bien un archivo de fragmentos dado (del cual no se conoce el archivo origen) solo los fragmentos. Para esto entre todos los miembros del grupo del curso se debe definir un estándar de archivo de entrada común para todos.

Su programa deberá listar los fragmentos cargados donde debe ser posible para el usuario ordenarlos y filtrarlos de múltiples formas: orden alfabético, largo de hilera, filtrar fragmentos mayores o menores a un valor dado, buscar palabras claves, etc.

Utilizando el algoritmo de ensamblaje de grafo de traslapes para la obtener la Superhilera Mínima Común, tomar los fragmentos daods y procesarlos para generar el grafo completo con todos los pesos de traslapes de todos los fragmentos dados. El usuario debe poder visualizar dicho grafo y explorarlo (navegando por él o visualizando todas sus partes: fragmentos y valores de traslapes (pesos) y la dirección de los arcos que indican el sufijo-prefijo correspondientes al traslape.

El usuario podría solicitar un grafo simplificado indicando el valor de traslape mínimo deseado. En este caso el programa deberá informar al usuario si el grafo es conexo o no.

Del grafo original su programa deberá generar el grafo conexo mínimo, es decir el que selecciona los mayores traslapes hasta que el grafo sea conexo. Adicionalmente el usuario podrá indicar un valor de traslape mínimo.

Con el grafo simplificado o no su programa deberá generar a partir de este y por orden de traslapes la hilera de salida como ensamblaje de los fragmentos dados.

Se deberá informar al usuario si el ensamblaje produjo o no "islas" de hileras, en el caso que el grafo de trabajo fuera no conexo por ejemplo o bien, no todas las hileras lograron la cobertura esperada.

Con este ensamblaje se debe reconstruir la hilera original siguiendo el algoritmo. Si tiene la hilera o archivo original deberá hacer una comparación de similitud entre la hilera original y la hilera reconstruida para su evaluación. Adicionalmente si no tenía los datos del archivo original su programa podría solicitarlo para validar su efectividad.

Análisis y Pruebas

Brinde múltiples corridas (experimentos) del programa para valorar su efectividad, proveyendo cambios en los parámetros usados para los fragmentos, para medir 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 #2 Ensamblaje – SusNombresCompletos
    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 Martes 28 de Noviembre antes de la media noche. Debe hacerlo de forma simultánea al correo earias@ic-itcr.ac.cr y copiarse usted mismo y su compañero de trabajo. No se aceptarán días de atraso para este proyecto.

7. Cualquier consulta puede hacerla al foro, o personalmente o al correo del profesor con copia al asistente al correo anterior.

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.