

Proyecto 1: Mapas Cromosómicos

Miércoles 24 de Septiembre

I. DESCRIPCIÓN

En este proyecto, Ud deberá desarrollar un algoritmo que produzca todos los posibles mapas cromosómicos dado que se conocen las probabilidades de los *crossovers* de varios pares de genes que podrían o no estar ligados. Toda la programación debe realizarse en C sobre Linux. El proyecto debe tener una interfaz gráfica apropiada. No se pueden cambiar las especificaciones de este documento. También, hay que escribir un reporte de temas relacionados a recombinación de genes.

II. PRELIMINARES

Cada grupo debe investigar algunos temas básicos biológicos y matemáticos asociados a mapas cromosómicos. Entre otras cosas, se sugiere que se estudien conceptos tales como genes ligados y genes no ligados, recombinación, centimorgans (cM), la relación entre cMs y pares de bases (distancia física), la relación entre cMs y la probabilidad de recombinación, la relación entre “mapas genéticos” y “mapas físicos”, etc. Es muy importante revisar los trabajos de Morgan y de Sturtevant, y analizar porque Mendel no reportó el fenómeno de la recombinación. Toda esta información debe recopilarse en la primera parte de un reporte escrito.

III. DATOS

La interacción con el usuario se hará por medio de interfaces gráficas que deben ser desarrolladas con GTK y Glade. El usuario proporcionará los siguientes datos:

- **Número de genes:** máximo de genes con los que se trabajarán. Se espera que este número sea tan grande como sea posible, pero se puede limitar para que la interfaz no resulte incómoda.
- **Identificación de los genes:** por defecto el programa generará un nombre genérico para cada gen (*e.g.*, G01, G02, etc.), pero el usuario puede modificar la descripción de algunos de ellos.
- **Probabilidad de *crossover*:** suponiendo que haya n genes se creará una tabla de $n \times n$ donde el usuario indicará la probabilidad de recombinación del gen i y el gen j . Esta tabla será simétrica por lo que la interfaz gráfica respectiva puede simplificarse. Nótese que el usuario podría no tener disponibles todas las probabilidades, pero esto no debe interpretarse como probabilidad 0.0 de *crossover*.

El programa debe ser capaz de grabar y leer esta información en y desde archivos que sigan el formato que cada grupo establezca.

	G01	G02	G03
G01	0.0	0.05	0.35
G02	—	0.0	0.02
G03	—	—	0.0

Tabla 1: Ejemplo de Datos Inconsistentes

IV. MAPAS CROMOSÓMICOS

Cuando el usuario lo solicite con algún botón de la interfaz, el programa producirá **todos los mapas cromosómicos posibles** que correspondan a los datos proporcionados hasta el momento por el usuario. Eventualmente, la mejor explicación podría colocar los genes en dos o más cromosomas separados, y a su vez podría haber varios mapas para cada uno de ellos. Estos mapas deben ser representados gráficamente, con una escala apropiada (esto es fundamental para que el proyecto tenga sentido). El usuario podrá hacer *zoom in* o *zoom out* de estos mapas. Sólo se deben considerar los pares de genes para los que haya información.

Trabajo extra opcional 1: los mapas están desplegados constantemente, pero se refrescan con cada dato nuevo de probabilidad que el usuario ingrese.

V. INCONSISTENCIAS E INFERENCIAS

Ciertos conjuntos de datos podrían ser inconsistentes o contradictorios. Por ejemplo, la **Tabla 1** muestra datos inconsistentes (¿Por qué?). El programa debe detectar estas situaciones u otras similares, reportarlas y negarse a generar el mapa genómico.

Similarmente, hay casos en los que, considerando los datos ingresados por el usuario hasta cierto punto, el programa podría calcular e inferir las probabilidades asociadas al *crossover* entre ciertos pares de genes no ingresados aún. Esta opción debe ser solicitada explícitamente por el usuario con algún botón. Por supuesto, puede que no haya nada aún que predecir.

VI. REPORTE

En la segunda parte del reporte escrito mencionado previamente, se debe mostrar su programa en ejecución (capturen pantallazos e inclúyanlos en sus reportes). Consulten libros de Biología Molecular donde aparezcan ejercicios de este tema que puedan ser resueltos con su programa. Aparte de los ejemplos que funcionen, analice aquellos casos especiales biológicos que no se ajusten a un modelo simple de generación de mapas cromosómicos.

VII. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto “no revisable” y recibe un 0 de calificación inmediata:

- La colaboración entre grupos se considera fraude académico.
- Todo el código debe estar escrito en C.
- Recuerde que usar IA para completar cualquier parte de su trabajo es fraude académico y tendrá muy serias consecuencias.
- El proyecto debe compilar y ejecutar en Linux. Todo debe estar **integrado**, explicaciones del tipo “*todo está bien pero no pudimos pegarlo*”¹ provocan la cancelación automática de la revisión.
- Para el momento de la revisión, su programa no debe tener “prints” de depuración. Luce muy poco profesional.
- Todas las interfaces deben ser gráficas, robustas y de mucha calidad.
- Se debe usar GTK y Glade. Es inaceptable hacer una interfaz de texto desplegada en un *widget*.
- No debe dar “Segmentation Fault” bajo ninguna circunstancia.
- La demostración debe hacerse en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales. No se puede levantar de un disco o memoria externa. No se pueden prestar máquinas entre grupos.

VIII. FECHA DE ENTREGA

Revisiones el **Miércoles 24 de septiembre** a las 11:30am en la oficina del profesor. Mande además un .tgz con todo lo necesario (fuentes, makefile, readme, etc.) a torresrojas.cursos.01@gmail.com. Ponga como subject: [BMC] Proyecto 1 – Fulano – Mengano – etc., donde Fulano, Mengano y etc. son los miembros del grupo.

¹esto incluye los supuestos casos cuando alguien del grupo de trabajo no hizo su parte – el profesor no está interesado en sus problemas de organización.