

Proyecto 0: Cruces Mendelianos

I. DESCRIPCIÓN GENERAL

Este programa hará cruces entre 2 progenitores siguiendo las leyes mendelianas de herencia. Cada progenitor tiene hasta 5 características binarias. Se mostrarán todos los descendientes posibles y la probabilidad de cada uno. La interfaz debe ser gráfica. Toda la programación debe realizarse en C sobre Linux. No se pueden cambiar las especificaciones de este documento.

II. ENTRADA Y SALIDA

La interacción con el usuario se hará por medio de interfaces gráficas que deben ser desarrolladas con GTK y Glade. El programa tendrá la posibilidad de grabar archivos con los datos particulares de cada problema ingresado por el usuario, para que puedan ser cargados de nuevo y editados si el usuario así lo desea. El formato de este archivo queda a discreción de cada grupo de trabajo. Se espera que, el día de la revisión, cada grupo cuente con **bastantes** archivos de pruebas para mostrar las capacidades de su proyecto.

III. CARACTERÍSTICAS

El primer dato solicitado es la cantidad n de características a considerar. Este será un número entre 1 y 5.

Trabajo extra opcional 1: Manejar $n > 5$

Cada característica será de naturaleza binaria con, obviamente, 2 estados excluyentes (*i.e.*, tallo alto, tallo bajo). Cada una será identificada con una sola letra del alfabeto que el usuario digitará (mayúscula para la forma dominante de la característica y la misma letra en minúscula para la forma recesiva). Por defecto se usarán las letras A, B, C, D, E. Sin embargo el usuario las puede cambiar a discreción (el cambio se puede hacer tanto en la letra minúscula como en la mayúscula y el programa actualizará automáticamente la otra letra). El usuario también proporcionará una descripción en texto de cada forma de la característica.

Por ejemplo:

- T: Tallo alto
- t: Tallo bajo

El programa debe ser robusto e impedir que se usen caracteres que no sean letras, que se repita una letra ya usada, o cualquier otro error en la entrada de estos datos.

IV. PADRES Y MADRES

Una vez que el usuario dé por terminada la entrada de características (con algún botón), el programa generará todos los **genotipos** posibles con las n características. Recuerde que cada característica tiene 2 alelos que a su vez pueden

ser dominantes o recesivos. Por ejemplo, si tuviéramos 2 características A y B, los 9 genotipos posibles son:

1. AABB
2. AABb
3. AAbb
4. AaBB
5. AaBb
6. Aabb
7. aaBB
8. aaBb
9. aabb

Nótese que “Aa” es lo mismo que “aA”.

Estos genotipos serán presentados al usuario como 2 colecciones idénticas pero separadas de botones de radio¹: cada colección corresponde a uno de los progenitores.

Se espera algún acomodo en tabla que sea agradable a la vista y que muestre al mismo tiempo las 2 colecciones de botones. Al flotar sobre estos botones se mostrará la burbujita de *tooltip* de dicho botón la cual describirá el **fenotipo** correspondiente al individuo (*e.g.*, podría haber 3 características y que cierto botón diga algo como “Tallo bajo, Flor morada, Semilla arrugada”).

El usuario presionará un botón de cada colección para establecer los 2 individuos que serán cruzados.

V. DESCENDENCIA

Cuando el usuario dé por terminada la selección de los 2 progenitores, se calcularán todos los descendientes posibles al separar los alelos de un progenitor y cruzarlos con los del otro. Este resultado se presentará en la forma de una tabla estándar de cruces mendelianos.

Dicha tabla mostrará todos los genotipos posibles de los descendientes, con colores para identificar todos los individuos “semejantes”. El usuario indicará si desea que los individuos sean coloreados por genotipo idéntico o por fenotipo idéntico.

Con un botón adicional se mostrarán los porcentajes que correspondan a cada fenotipo idéntico y los porcentajes que correspondan a cada genotipo idéntico. En ambos casos los datos se presentarán ordenados de mayor a menor.

VI. REQUISITOS INDISPENSABLES

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

- Todo el código debe estar escrito en C
- El proyecto debe compilar y ejecutar en Linux

¹este widget sólo permite un botón activo en cada colección

- Todas las interfaces deben ser gráficas
- Se debe usar GTK y Glade
- No debe dar “Segmentation Fault” bajo ninguna circunstancia.

Trabajo extra opcional 2: hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.

VII. FECHA DE ENTREGA

Revisiones en clase el **Miércoles 15 de Marzo**. Mande además un .tgz con todo lo necesario (fuentes, makefile, readme, etc.) a torresrojas.cursos@gmail.com. Ponga como subject: BMC - Proyecto 0 - Fulano - Mengano, donde Fulano y Mengano son los 2 miembros del grupo.