

Trabajo práctico final: MutaViz

1 Objetivo	1
2 Contexto	1
3 Requerimientos detallados	2
3.1 Carga de secuencias	3
3.2 Mutación	3
3.2.1 Modo Manual	3
3.2.2 Modo Automático	3
3.3 Visualización	4
3.4 Tecnologías	4
4 Forma de entrega	4

1 Objetivo

El objetivo de este trabajo práctico es integrar los conceptos biológicos y bioinformáticos desarrollados durante la materia, junto con los conocimientos, prácticas y habilidades propias de la informática y la programación adquiridas en otras materias, a través de la construcción de un software simple pero innovador que sirva de asistencia al proceso del análisis bioinformático y/o herramienta educativa para la enseñanza de la Biología.



2 Contexto

Como aprendimos durante el desarrollo de la materia, los organismos pueden acumular cambios heredables (en su ADN) con el paso del tiempo. Estos cambios a nivel genético, se traducen en cambios a nivel de las proteínas para las cuales estos genes codifican. Como ya hemos visto en clase, también, las proteínas presentan estructura primaria, secundaria, terciaria y cuaternaria. Como hemos visto en clase la estructura de las proteínas se encuentra estrechamente relacionada a la función ([Todd et al. 2001](#)). Así mismo, teniendo en cuenta los conceptos relativos a la evolución, no les resultará ilógico pensar que: dado que la forma en que el plegamiento de una proteína se encuentra dirigida por su secuencia, y que está a su vez es codificada por la secuencia de un gen, entonces existirán restricciones evolutivas para los posibles cambios en la secuencia y estructura proteica ([Zea et al. 2018](#); [Grishin 2001](#)). Es decir, que no todas las posibles mutaciones a nivel genético serán viables o producirán una proteína funcional, que le permita al organismo sobrevivir ([Strokach et al. 2019](#); [Bujnicki 2001](#)). Es, por tanto, importante comprender y predecir los cambios estructurales que una mutación genética introduce ([Bueno et al. 2018](#)).

Para pensar: ¿qué factores pueden hacer que las mutaciones ocurran? ¿Cómo se relacionan estas con la diversidad biológica?

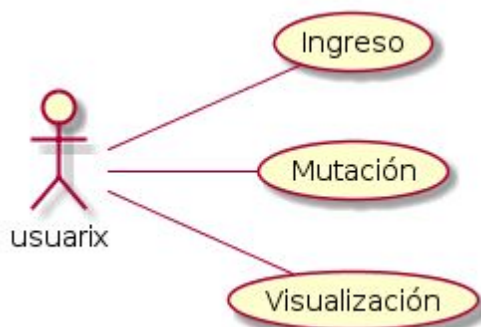
Estudiar los cambios estructurales que producen las distintas mutaciones es de particular importancia para la bioinformática, ya que nos permite predecir y manipular estos cambios con fines biotecnológicos ([Gerlt 1987](#)). La técnica de mutagénesis dirigida, es decir la inducción de ciertas mutaciones en ciertas posiciones de la proteína para potenciar su actividad o inducir nuevas funciones, es una herramienta de ingeniería genética muy frecuentemente utilizada. En este sentido la construcción de herramientas que nos permitan visualizar y predecir los cambios estructurales que una mutación induce, ayudaría a mejorar dichas aplicaciones.

En este contexto nos proponemos desarrollar una aplicación que nos permita predecir el plegado proteico desde una secuencia de ácidos nucleicos y visualizar los cambios que está sufriendo al introducirse mutaciones. Si bien ya existen programas capaces de dar respuesta parciales a algunos de estos cuestionamientos, no existen herramientas interactivas que las integren, permitiendo al usuario o usuaria visualizar estos cambios de forma dinámica. Así mismo, la herramienta permitirá seleccionar el idioma (inglés/español), haciendo de esta una herramienta educativa de gran importancia, que podría ser aplicada al aprendizaje-enseñanza de la Biología, ya que no existen herramientas de estas características en español.

Para pensar: ¿nos podría servir este software para establecer relaciones secuencia-estructura-función en proteínas homólogas?

3 Requerimientos detallados

El objetivo es desarrollar un software que permita resolver tareas de análisis de proteínas. El software podrá ser desarrollado bajo cualquier arquitectura y empleando cualquier tecnología de **código libre**, y deberá dar respuesta a los casos de uso expresados a continuación.



3.1 Carga de secuencias

El sistema debe permitir ingresar una secuencia en el formato FASTA¹, para su posterior procesamiento. Si la secuencia ingresada es inválida, el software deberá notificarlo apropiadamente, mostrando a el/la usuario/a un mensaje claro y útil.

3.2 Mutación

El sistema debe poder simular mutaciones sobre la secuencia anteriormente ingresada, soportando dos **modos** de trabajo diferente: **manual** y **automático**. Una vez ingresada una secuencia correcta, el software deberá pedirle a el/la usuario/a que elija entre uno de los modos.

3.2.1 Modo Manual

El modo manual permite a los/las usuarios/as mutar manualmente la secuencia: se mostrará la secuencia por pantalla, y se podrán modificar individualmente cada una de las bases, como así también agregar nuevas o quitar existentes. Dado que no todas las letras representan bases válidas, es fundamental que el software valide (durante o al terminar la edición, a criterio del equipo de desarrollo) que la cadena resultante sea válida.

3.2.2 Modo Automático

El modo automático permite mutar la secuencia de forma no supervisada, siguiendo alguno de los siguientes modos:

- Azar: podrá suceder de forma equiprobable una **modificación** de una base. El software deberá permitir ingresar la cantidad de mutaciones que ocurrirán.
- Permite ajustar la probabilidad de ocurrencia de alguno de los tres tipos de mutaciones, para que no sean necesariamente equiprobables. Siguiendo distintos modelos evolutivos, que ponderen los cambios de distinto modo (ver teoría de inferencias evolutivas):
 - Jukes - Cantor: pondera de igual modo transiciones y transversiones (equiprobables).
 - Kimura 2 parámetros: igual frecuencia de bases, pero distinta probabilidad entre transversiones y transiciones.

Tanto el modelo evolutivo como la cantidad de mutaciones que se realizarán son seleccionables por el/la usuario/a.

¹ https://es.wikipedia.org/wiki/Formato_FASTA

3.3 Visualización

Finalmente, el sistema deberá permitir visualizar proteínas (preferentemente alineadas o superpuestas), en dos momentos diferentes:

- tras la carga de la secuencia
- tras la mutación de la secuencia

3.4 Tecnologías

El software podrá estar implementado utilizando cualquier tecnología. Sin embargo, se recomienda fuertemente utilizar alguna de las siguientes:

- Python
- JavaScript / TypeScript

Para las tecnologías de visualización, a continuación se mencionan algunas opciones:

- <https://github.com/McMenemy/viztein>
- <https://github.com/arose/ngl>
- <https://github.com/jowoojun/biovec>
- <https://github.com/intermine/bluegenes-protein-visualizer>
- <https://pymol.org/2/>
- <https://pypi.org/project/backmap/>

No todas de igual calidad y presentan características diferentes, además de que tampoco son las únicas opciones disponibles. Queda a criterio del equipo la elección de la biblioteca.

El software no debe estar desarrollado bajo ninguna arquitectura particular, pudiendo usarse indistintamente cualquiera de las siguientes:

- *Desktop*
- Web MVC del lado del servidor
- Web MVC del lado del cliente

4 Forma de entrega

El trabajo práctico se realizará en equipos de hasta 4 integrantes, y se entregará el día 4 de Diciembre. El mismo deberá además estar o a un repositorio público en Github (creado para la materia), el cual deberá contener un archivo README.md con los datos de I@s integrantes del equipo. Los trabajos serán presentados por los miembros del equipo en una exposición oral (con DEMO), la última clase de la cursada.