

LABORATORIO 5

En esta práctica, se aprende:

- **Realizar un control de versiones:** este es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar versiones específicas más adelante. Con este sistema se puede volver a versiones anteriores de tus archivos; regresar a una versión anterior del proyecto completo; comparar cambios a lo largo del tiempo; ver quién modificó el archivo, entre otras muchas cosas. Además, realizando un control de versiones se puede tener la seguridad de poder recuperar archivos perdidos, ya que cada copia de un repositorio es una copia de seguridad completa de un proyecto y su Git historial.
El control de versiones se consigue con el manejo de los **comandos git** desde la terminal de Mobaxterm. Entre los más usados están: **git add**, para añadir los cambios realizados en los distintos ficheros o directorios; **git status**, sirve para mostrar los diferentes estados de los archivos en tu directorio de trabajo y área de ensayo, es decir, qué archivos están modificados y sin seguimiento y cuáles con seguimiento, pero no confirmados aún; **git commit**, sirve para introducir un mensaje de confirmación de los cambios introducidos; **git push**, para sincronizar tu repositorio local con tu repositorio en GitHub y poder sincronizar los cambios realizados.
- **Abrir una nueva línea de desarrollo** en el repositorio local, gracias al comando **git branch** (nombre de la nueva rama), y posteriormente sincronizar estos cambios con el repositorio remoto.
- **Resolución de conflictos** que se generan cuando varias personas trabajan sobre un mismo repositorio y no se sabe cuál es el archivo bueno y cuál la copia.
- Usar **GitHub**, que es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código.
- Por último, esta práctica también sirve para repasar conceptos sobre los **Shell scripts** y el uso de **SLURM** aprendidos en otros laboratorios.

Como **aspecto negativo** de esta práctica me gustaría comentar que en el **apartado 1** de la parte avanzada, cuando intentaba ejecutar el archivo **filecut.sh**, que corta los 4 ficheros **Sample.fastq** en 25 trozos y se queda con el primero, dentro del fichero **submit_lab5_ampere-alumno25.sh**, que realiza el mismo procedimiento pero en paralelo, no se ejecutaba correctamente. Por tanto, decidí ejecutar por separado el archivo **filecut.sh** y el **submit_lab5_ampere-alumno25.sh** y comprobar que ambos realizan lo mismo. Por eso, en mi repositorio de GitHub, tengo archivos denominados **Partes** que son los que he obtenido con el programa **filecut.sh** y los **Sample.fast** que son los que he utilizado para actualizar estos ficheros.

Como **conclusión** de esta práctica, obtengo que el control de versiones es útil, entre otras cosas, para ayudar a controlar y compartir el código de trabajo científico de los investigadores. Además, otra ventaja bastante importante es poder sincronizar tu repositorio local con tu GitHub, a través del uso de distintos comandos haciendo así que el control de archivos sea menos pesado.