Usando git con R y Rstudio

Cada vez es más habitual que en los cursos se distribuyan o recomienden materiales que no son archivos sino repositorios de github. De hecho, después de este mensaje os facilitaré los enlaces a materiales del curso que se encuentran en repositorios de github.

Hay diversos motivos para proceder así

- Si deseamos que un ejemplo, un caso, o un curso, se pueda reproducir necesitamos, aparte del documento que lo describe, el código de R, y quizás otros archivos como imágenes, archivos CEL o archivos de covariables ("targets").
- Una manera "natural" de organizar esto es guardarlo todo en el mismo directorio. RStudio promueve esta forma de trabajar a través del concepto de "proyecto". Creando un proyecto de Rstudio estamos encapsulando en un directorio todo lo relacionado con el mismo.
- Siempre es posible crear un proyecto, aunque sólo vayamos a hacer un pequeño análisis. De hecho, mi recomendación es que lo hagáis siempre, porqué es una forma cómoda de asegurarse que todo lo relacionado con el análisis permanece junto.

- Otra forma de organizar los datos en un directorio es poniéndolo "bajo control de versión" es decir almacenarlo en algún sitio, local o remoto, y utilizar un programa como "git" o "SVN" que permite mantener un historial de cambios. Hoy en día se ha popularizado enormemente el uso combinado de git y de github.
- git es un programa de control de versiones
- github es un servidor web en donde pueden alojarse proyectos bajo control de versión que denominamos "repositorios".

- Y aquí es donde la cosa se pone interesante. Es posible crear un proyecto de Rstudio asociándolo con un repositorio de github. En concreto esto se hace clonando el repositorio a un proyecto de Rstudio. Una vez hecha esta asociación podemos disfrutar de las ventajas de los dos sistemas
- Por un lado, tenemos todo nuestro proyecto encapsulado al tenerlo como "proyecto de Rstudio"

- Por otro lado, lo tenemos bajo control de versión con git en github. Esto significa que podemos
    - Si se trata de un repositorio propiedad de otra persona podemos mantenerlo actualizado fácilmente incorporando con facilidad los cambios que se vayan produciendo.
    - Si tenemos derechos de escritura en el repositorio, podemos actualizar nuestros cambios directamente desde Rstudio.

Podéis ampliar esta breve explicación mediante alguno de los muchos materiales que podéis encontrar por internet.

En castellano

- Curso breve: El control de versiones con Git : https://swcarpentry.github.io/git-novice-es/
- Taller sobre paquetes de R y github. Obsérvese que el taller sobre github es él mismo un repositorio de github

  https://github.com/alexsanchezpla/2017-03-Rpackages_and_Github

El uso de github puede parecer extraño al principio y de hecho suele costar dominarlo. Sin embargo, cuando uno se acostumbra a utilizarlo sucede lo mismo que con Rmarkdown: no hacerlo así nos parece una forma basta de trabajar porque nuestros proyectos no son reproducibles.

No es exagerado decir que hoy en día la gran mayoría de la comunidad bioinformática utiliza github -o algo parecido para el control de versiones- y Rmarkdown o algo parecido para garantizar la reproducibilidad de los proyectos. Como esto es muy importante iremos volviendo sobre ello y nos lo tomaremos con calma. Pero tened en cuenta una cosa: es un camino del que no hay vuelta atrás :-)


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.