General information about git

Contents

Git is a successful version control software (see [Wikipedia](#) for detailed information). It was created in 2005 and got rapidly adopted by software developers, especially because it is very fast and scalable. Its functions allow for parallel development and maintenance of large projects, like linux development.

Why everyone is using git

While developed for software, Git has been used for many different kind of projects and platforms such as GitHub, GitLab, Gogs, GitLea and others. These platforms brought project management tools into the Git workflow, facilitating community building around projects like the Turing Way book. Sometimes, users of these platform do not even know about Git.

Git Limitations

Git is not magic and it is good to know about its limitations. Especially, Git works best with small text files. Git starts to be impracticable when too many files are present, or when the repository becomes too big (1 TB is about the limit). As a Git repository stores every version of every file that is added to it, large files that undergo regular modifications can inflate the size of a·project significantly. In research projects, datasets often contains thousands of files and/or contain (very) large files. While one can use Git for non-software application, one needs to plan to use specific workflows and/or additional tools to be able to use Git tools. It is particularly problematic because everything will work fine in the beginning and it is very difficult to solve issues when the project is ongoing. It is therefore important to plan ahead, and try to avoid big repositories. For instance, one can split the

files into different repositories and save binary files outside of Git. There are tools allowing that while keeping git at the core of the version control (git-annex and submodules are possible technologies, see section [data version control](#) and [research projects](#)).

How one uses Git

If you ever made modification of files in GitHub, you have probably used Git without even realising it. When you push the `commit changes` button on a Git platform, Git was acting in the background to `add` the changes to the index, `commit` them with a message, and push it to the repository.

Several software allows to use Git without using the command line, locally. Here is a non-exhaustive list of software one can also use, please refer to their own documentation:

- [RStudio](#)
- Sourcetree
- Gitkracken
- [Visual Studio Code](#)

In many cases, one still needs to use the command lines for complex matters, and we present the main Git functions usage in this book (see [Getting Started with Git](#)).