LEIDEN UNIVERSITY MEDICAL CENTER

Analysis projects skeleton

Jeroen F. J. Laros

Leiden Genome Technology Center

Department of Human Genetics

Center for Human and Clinical Genetics

Introduction

Shared projects.

Most of us work on multiple projects with multiple people.

That is why is is convenient to:

- Have everything in one place.
  - Data.
  - Code.
  - Documentation.

Introduction

Shared projects.

Most of us work on multiple projects with multiple people.

That is why is is convenient to:

- Have everything in one place.
  - Data.
  - Code.
  - Documentation.
- Have the same structure for all projects.

Starting a project

Project skeleton.

Usage:

- Make a *fork* (copy) of the skeleton project.
- Rename the project.
- Clone your project.
- Start working with it.

https://git.lumc.nl/lgtc-bioinformatics/project-skeleton

Starting a project

Project skeleton.

Usage:

- Make a *fork* (copy) of the skeleton project.
- Rename the project.
- Clone your project.
- Start working with it.

Configure your project.

- Choose to make your project public or not.
  - Public by default.
  - Public really means public.
- Add the people that work on this project.

https://git.lumc.nl/lgtc-bioinformatics/project-skeleton

Project structure

Global overview.

Project layout:

- analysis
- data
- doc
- src

Ideally, every directory in the project has a **`README`** file.

Project structure

The toplevel `README` file.

This file contains general information about the project, for example:

- Who leads the project.
- Who participates in the project.
- The amount of hours people have spent on this project.

Project structure

The `doc` directory.

Documentation on the project:

- Annotation of the data.
- Goal of the project.
- Related work and literature.
    - You may want to note who provided the documentation.

Project structure

The `data` directory.

Used to store all raw data.

The **`README`** contains:

- Description of the delivered data.
  - Sequencing centre.
  - Platform.
  - Molecular type.
  - Owner.
  - Gatherer.
- Description of other data.
  - Perhaps you already got BAM files.
    - Who aligned it?
    - Which aligner?

Project structure

The `analysis` directory.

All analysis related files are stored here:

- Run scripts.
- Make files.
- Result files.

Try to separate self-contained parts of the analysis in their own subdirectories and document dependencies in a **`README`** file.

- Normal data analysis.
- k-mer analysis.

Project structure

The `src` directory.

Any custom scripts and specific software versions for this project.

When these scripts are useful for other projects, move them to their own repository.

Working with large files

Git is not designed for massive files.

Some problems with large files:

- Limited storage on the server.
- Checking out a repository would take a long time.

It also does not make much sense:

- These files are usually *static*.
- And probably *binary*.

http://git-annex.branchable.com/

Working with large files

Git is not designed for massive files.

Some problems with large files:

- Limited storage on the server.
- Checking out a repository would take a long time.

It also does not make much sense:

- These files are usually *static*.
- And probably *binary*.

We do want to have some way to track our input and output data. This can be done with **`git-annex`**.

http://git-annex.branchable.com/

Working with large files

Git annex.

Manage files with git, without checking their contents in.

- Manage large files without storing them.
- Store file checksums.
- Prevent files from being deleted accidentally.

Working with large files

Git annex.

Manage files with git, without checking their contents in.

- Manage large files without storing them.
- Store file checksums.
- Prevent files from being deleted accidentally.

You first have to enable this for your repository.

```
1   $ git annex init "<name>"
```

Listing 1 : Enable git-annex.

Working with large files

Adding big files.

In our master repository, we annex a file.

```
1   $ git annex add <filename>
2   $ git commit
```

Listing 2 : Adding files.

Working with large files

Adding big files.

In our master repository, we annex a file.

```
1  $ git annex add <filename>
2  $ git commit
```

Listing 2 : Adding files.

In a clone, this file will visible, but not really present.

```
1  $ file <filename>
2  <filename>: broken symbolic link to ...
3  $ git annex get <filename>
```

Listing 3 : Make a file available.

Working with large files

Removing files.

As long as there are enough copies available, you can remove files.

```
1  $ git annex drop <filename>
2  drop bigfile (unsafe)
3  git-annex: drop: 1 failed
```

Listing 4 : A failing drop command.

Working with large files

Removing files.

As long as there are enough copies available, you can remove files.

```
1  $ git annex drop <filename>
2  drop bigfile (unsafe)
3  git-annex: drop: 1 failed
```

Listing 4 : A failing drop command.

It is actually quite well protected.

```
1  $ rm -rf <repository>
2  rm: cannot remove <repository>/.git/annex/objects/...
```

Listing 5 : rm fails too.

Working with large files

Synchronise your results.

Let the other repositories know what you have done.

```
1   $ git annex sync
```

Listing 6 : Synchronise with all repositories.

Working with large files

Synchronise your results.

Let the other repositories know what you have done.

```
1   $ git annex sync
```

Listing 6 : Synchronise with all repositories.

You can choose to sync with a selection of repositories.

```
1   $ git annex sync origin
```

Listing 7 : Synchronise with a selection.

Working with large files

Working together on the same clone.

Sometimes you need to work with other people on the same repository clone.

- Where the large files are stored.

Use the following command to give group access:

```
1   $ find -type d -exec chmod 775 {} \;
2   $ find -type f -exec chmod 664 {} \;
```

Listing 8 : Make everyting group writable.

Questions?

Acknowledgements:

Martijn Vermaat
Zuotian Tatum

http://git-annex.branchable.com/

https://git.lumc.nl/lgtc-bioinformatics/project-skeleton