

Concept of Git and git-based platforms

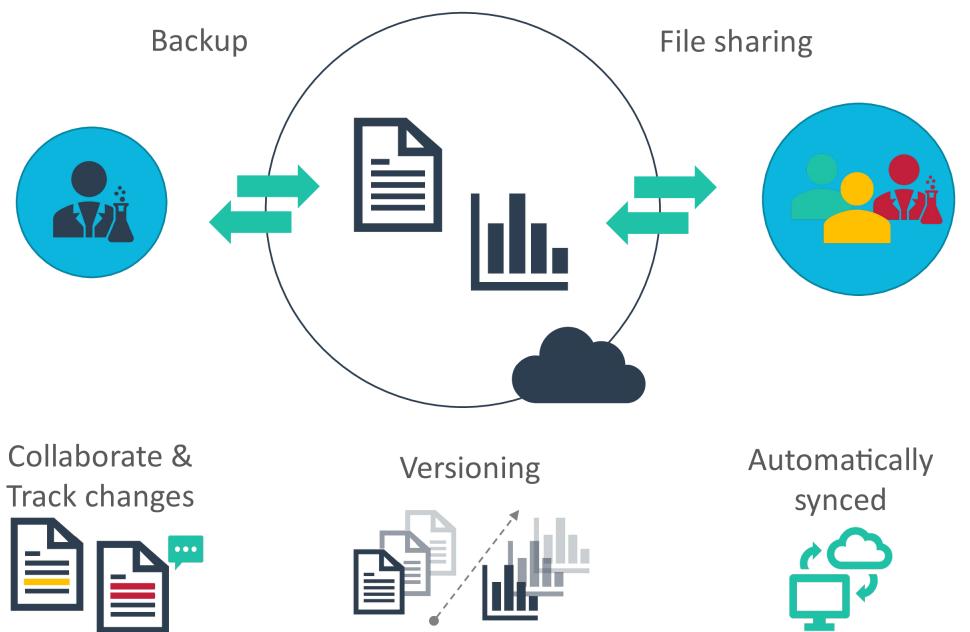
Cloud Services

- ✓ Documents
- ✓ Small data
- ✓ Presentations

X Code

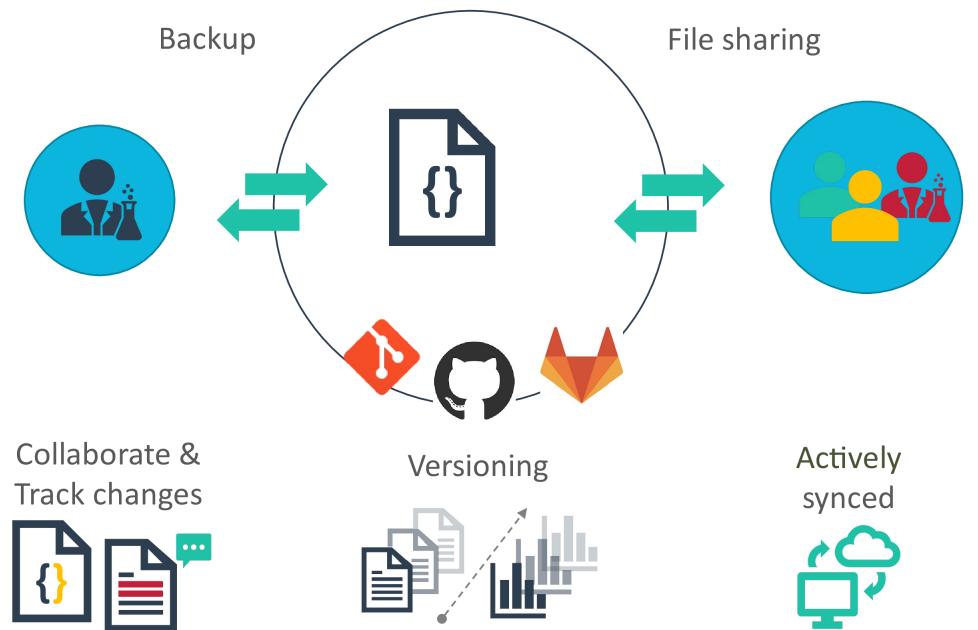
X Data analytical projects

X Big (“raw”) data



Git and git platforms

- ~ Documents
- ✓ Small data
- ~ Presentations
- ✓✓ Code
- ✓✓ Data analytical projects
- ~ Big (“raw”) data



Why git? => Why code?

- Save time
- Avoid doing repetitive tasks “by hand”
- Reuse scripts, analyses, pipelines
- Reproduce results

A simple example: RNASeq project

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   └── barley_sample03.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```

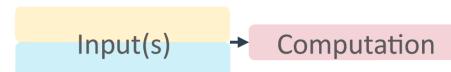
A simple example: RNASeq project

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   └── barley_sample03.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```

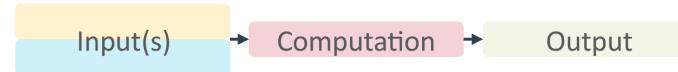
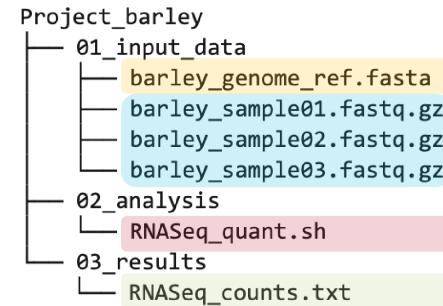
Input(s)

A simple example: RNASeq project

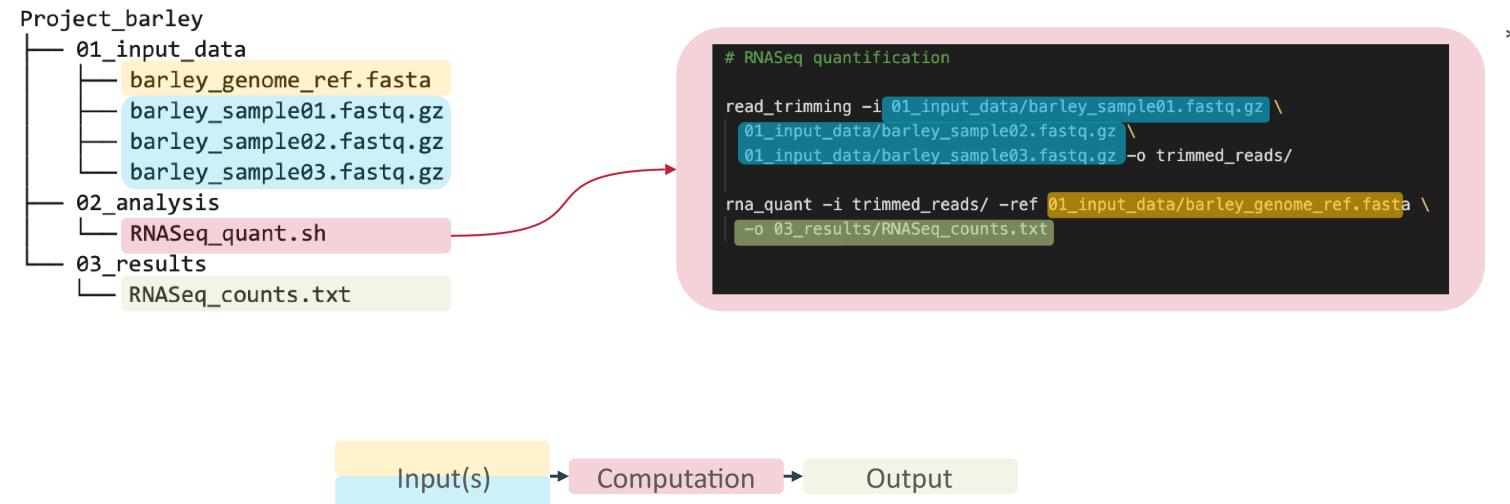
```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   └── barley_sample03.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```



A simple example: RNASeq project

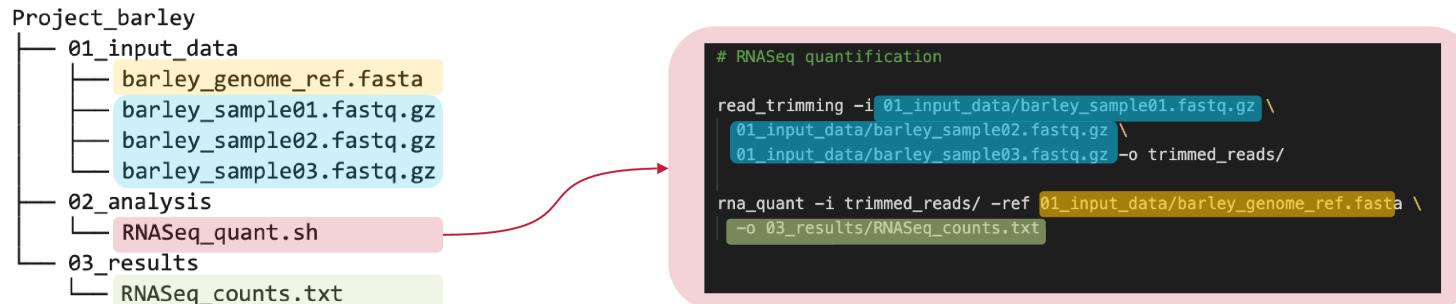


A simple example: RNASeq project



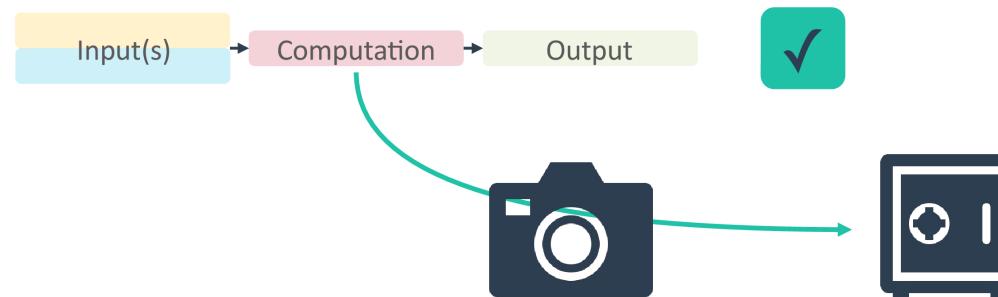
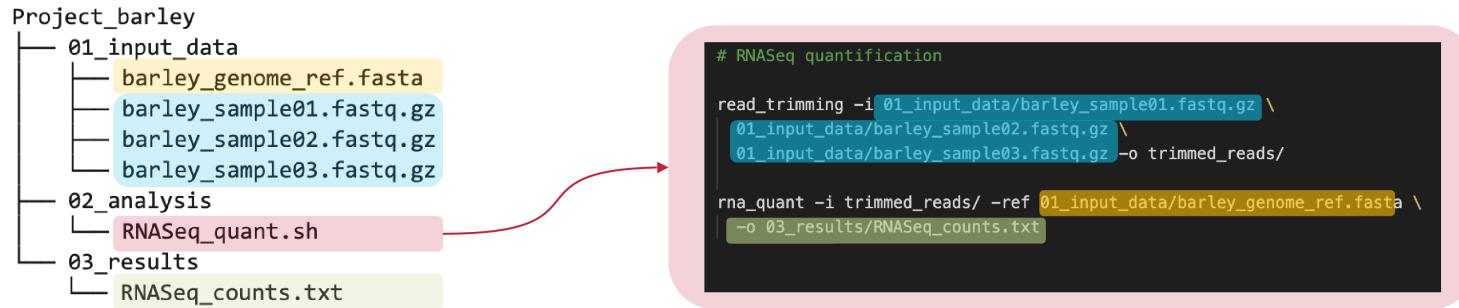
Take snapshots of your code work...

(... as long as it works)



Take snapshots of your code work...

(... as long as it works)



Scenario 1: More data

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   └── barley_sample03.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```

```
# RNASeq quantification

read_trimming -i 01_input_data/barley_sample01.fastq.gz \
01_input_data/barley_sample02.fastq.gz \
01_input_data/barley_sample03.fastq.gz -o trimmed_reads/
rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
-o 03_results/RNASeq_counts.txt
```

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   ├── barley_sample03.fastq.gz
│   ├── barley_sample04.fastq.gz
│   ├── barley_sample05.fastq.gz
│   └── barley_sample06.fastq.gz
```

Scenario 1: More data

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   └── barley_sample03.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   ├── barley_sample03.fastq.gz
│   ├── barley_sample04.fastq.gz
│   ├── barley_sample05.fastq.gz
│   └── barley_sample06.fastq.gz
```

```
# RNASeq quantification

read_trimming -i 01_input_data/barley_sample01.fastq.gz \
01_input_data/barley_sample02.fastq.gz \
01_input_data/barley_sample03.fastq.gz -o trimmed_reads/
rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
-o 03_results/RNASeq_counts.txt
```

```
# RNASeq quantification

read_trimming -i 01_input_data/barley_sample01.fastq.gz \
01_input_data/barley_sample02.fastq.gz \
01_input_data/barley_sample03.fastq.gz \
01_input_data/barley_sample04.fastq.gz \
01_input_data/barley_sample05.fastq.gz \
01_input_data/barley_sample06.fastq.gz -o trimmed_reads/
rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
-o 03_results/RNASeq_counts.txt
```

Scenario 1: More data

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   └── barley_sample03.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   ├── barley_sample03.fastq.gz
│   ├── barley_sample04.fastq.gz
│   ├── barley_sample05.fastq.gz
│   └── barley_sample06.fastq.gz
└── 02_analysis
    ├── RNASeq_quant.sh
    ├── RNASeq_quant_first_samples.sh
    ├── RNASeq_quant_including_all_samples.sh
    ├── RNASeq_quant_including_all_samples_updated.sh
    └── RNASeq_quant_including_all_samples_updated_v2.sh
└── 03_results
    └── RNASeq_counts.txt
```

```
# RNASeq quantification

read_trimming -i 01_input_data/barley_sample01.fastq.gz \
01_input_data/barley_sample02.fastq.gz \
01_input_data/barley_sample03.fastq.gz -o trimmed_reads/
rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
-o 03_results/RNASeq_counts.txt
```

```
# RNASeq quantification

read_trimming -i 01_input_data/barley_sample01.fastq.gz \
01_input_data/barley_sample02.fastq.gz \
01_input_data/barley_sample03.fastq.gz \
01_input_data/barley_sample04.fastq.gz \
01_input_data/barley_sample05.fastq.gz \
01_input_data/barley_sample06.fastq.gz -o trimmed_reads/
rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
-o 03_results/RNASeq_counts.txt
```

Let git track changes and keep things clean

```
Project_barley
└── 01_input_data
    ├── barley_genome_ref.fasta
    ├── barley_sample01.fastq.gz
    ├── barley_sample02.fastq.gz
    ├── barley_sample03.fastq.gz
    ├── barley_sample04.fastq.gz
    ├── barley_sample05.fastq.gz
    └── barley_sample06.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```

```
Project_barley > 02_analysis > $ RNASeq_quant.sh
1 # RNASeq quantification
2
3 read_trimming -i 01_input_data/barley_sample01.fastq.gz \
4 01_input_data/barley_sample02.fastq.gz \
5 01_input_data/barley_sample03.fastq.gz -o trimmed_reads/
6
7 rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
8 -o 03_results/RNASeq_counts.txt
9
10
11
12
13
14
```

“version 1”

```
1 # RNASeq quantification
2
3 read_trimming -i 01_input_data/barley_sample01.fastq.gz \
4 01_input_data/barley_sample02.fastq.gz \
5+ 01_input_data/barley_sample03.fastq.gz \
6+ 01_input_data/barley_sample04.fastq.gz \
7+ 01_input_data/barley_sample05.fastq.gz \
8+ 01_input_data/barley_sample06.fastq.gz -o trimmed_reads/
9
10 rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
11 -o 03_results/RNASeq_counts.txt
12
13
14
```

“version 2”

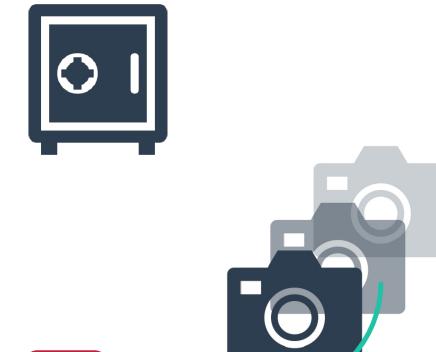
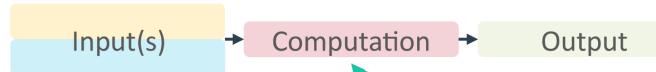
Scenario 2: Pipeline breaks

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   ├── barley_sample03.fastq.gz
│   ├── barley_sample04.fastq.gz
│   ├── barley_sample05.fastq.gz
│   └── barley_sample06.fastq.gz
├── 02_analysis
└── 03_results
    └── RNASeq_counts.txt
```



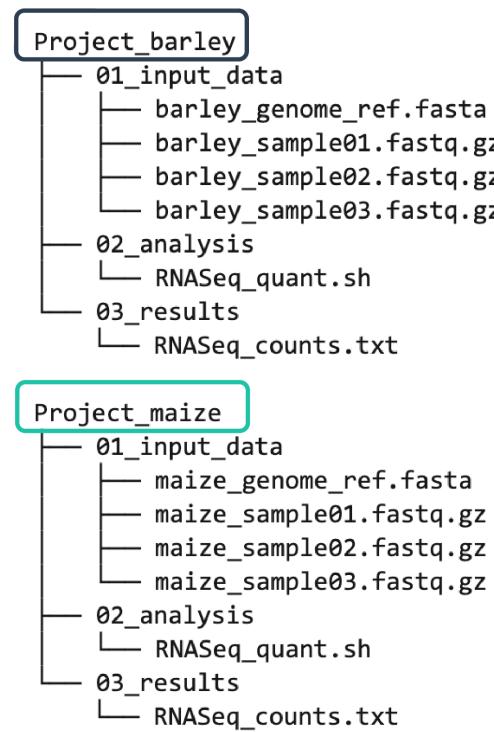
Revert to snapshot

```
Project_barley
├── 01_input_data
│   ├── barley_genome_ref.fasta
│   ├── barley_sample01.fastq.gz
│   ├── barley_sample02.fastq.gz
│   ├── barley_sample03.fastq.gz
│   ├── barley_sample04.fastq.gz
│   ├── barley_sample05.fastq.gz
│   └── barley_sample06.fastq.gz
└── 02_analysis
    └── RNASeq_quant.sh
└── 03_results
    └── RNASeq_counts.txt
```

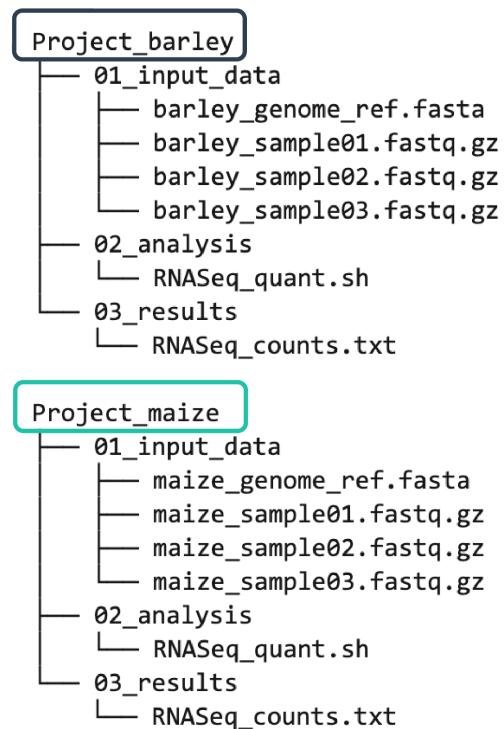


“version xy”

Scenario 3: New project, same type of data and analysis



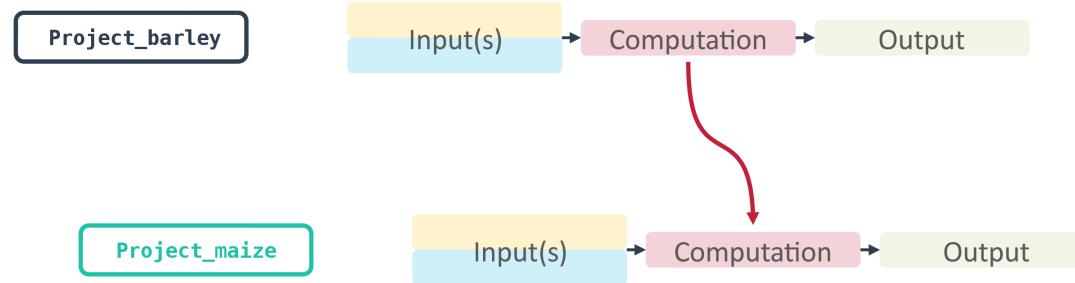
Scenario 3: New project, same type of data and analysis



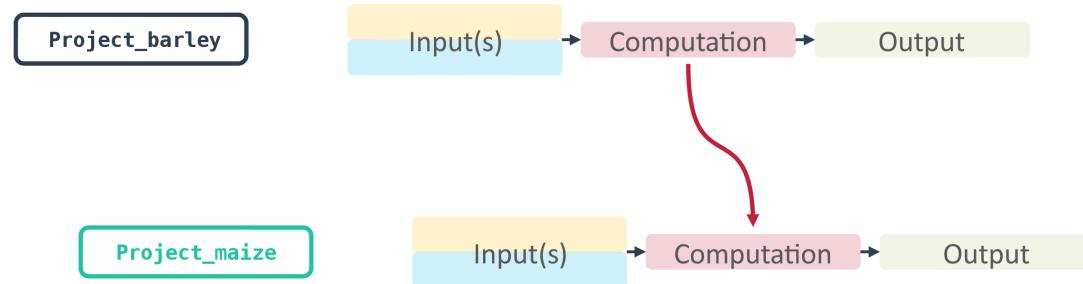
```
# RNASeq quantification
read_trimming -i 01_input_data/barley_sample01.fastq.gz \
01_input_data/barley_sample02.fastq.gz \
01_input_data/barley_sample03.fastq.gz -o trimmed_reads/
rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
-o 03_results/RNASeq_counts.txt
```

```
# RNASeq quantification
read_trimming -i 01_input_data/maize_sample01.fastq.gz \
01_input_data/maize_sample02.fastq.gz 01_input_data/maize_sample03.fastq.gz \
-o trimmed_reads/
rna_quant -i trimmed_reads/ -ref 01_input_data/ maize_genome_ref.fasta \
-o 03_results/RNASeq_counts.txt
```

Re-use code



Re-use code



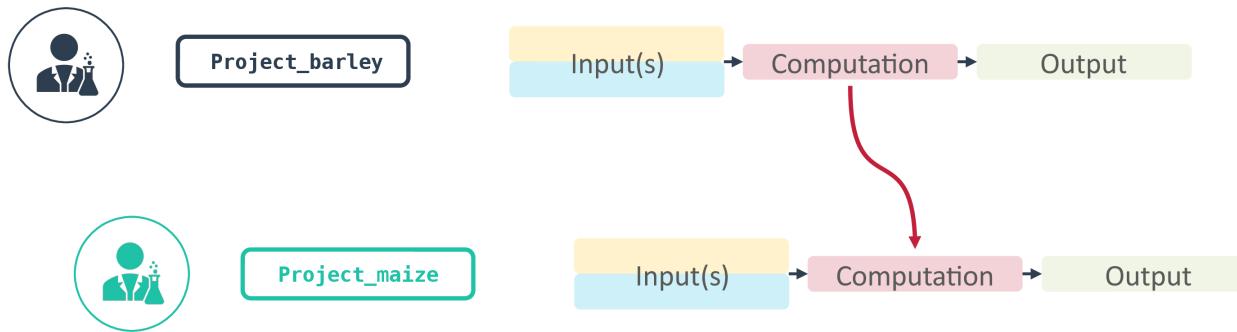
```
1 # RNASeq quantification
2
3 read_trimming -i 01_input_data/barley_sample01.fastq.gz \
4 01_input_data/barley_sample02.fastq.gz \
5 01_input_data/barley_sample03.fastq.gz \
6 -o trimmed_reads/
7
8 rna_quant -i trimmed_reads/ -ref 01_input_data/barley_genome_ref.fasta \
9 -o 03_results/RNASeq_counts.txt
10
```

“version barley”

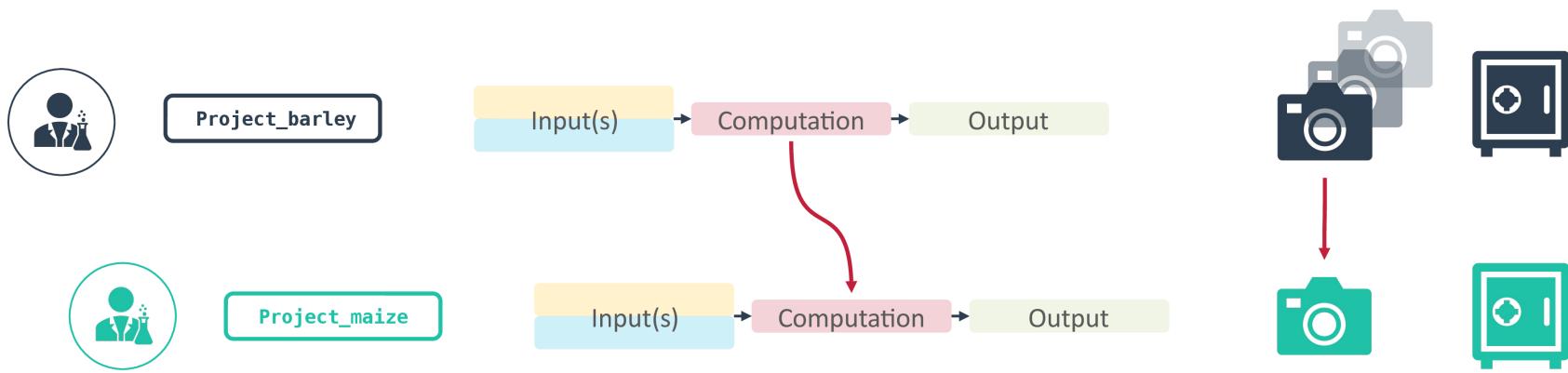
```
1 # RNASeq quantification
2
3+ read_trimming -i 01_input_data/maize_sample01.fastq.gz \
4+ 01_input_data/maize_sample02.fastq.gz 01_input_data/maize_sample03.fastq.gz
5+ -o trimmed_reads/
6
7+ rna_quant -i trimmed_reads/ -ref 01_input_data/ maize_genome_ref.fasta -o 03_results/RNASeq_counts.txt
8
9
10
```

“version maize”

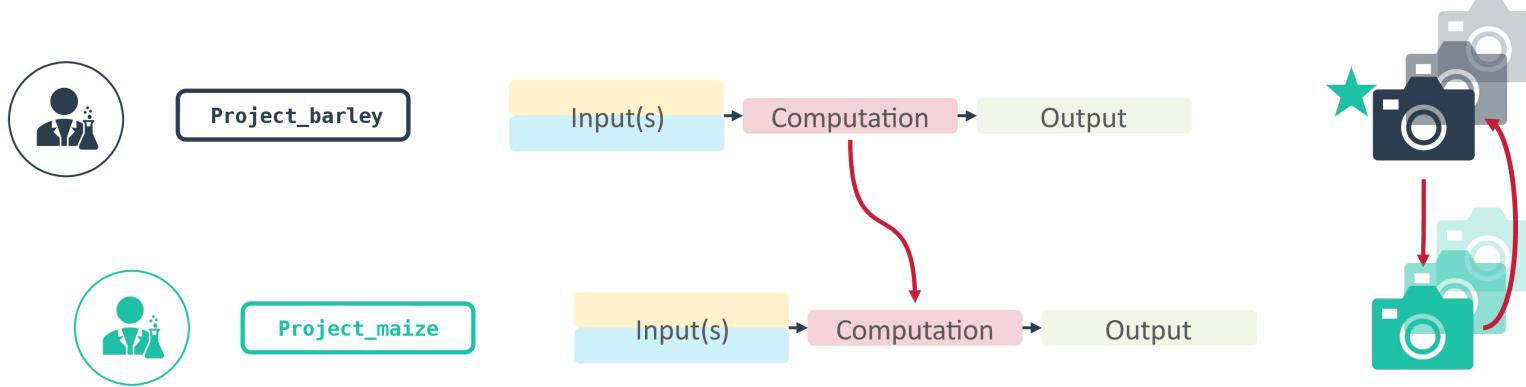
Re-use code – People have done this



Re-use code – People have done this



Re-use code – Link and contribute



Git: summary

- Version control system
- Git “repository” = a central data package (directory)
- Allows to track changes to any file in the repository
 - **What** was changed
 - **When** was it changed
 - **By whom** was it changed
 - **Why** was it changed?

GitHub and GitLab

- A well-documented cloud environment
- Active syncing
- Not automatically synced
- Non-automated version control
- You have the control what changes to track and what to sync
- Time machine to go back to older versions

GitHub and Gitlab team projects

Simplifies concurrent work & merging changes

- Online service to host our projects
- Share code with other developers
- Others can download our projects, work on and contribute to them
- They can upload their changes and merge them with the main project

Cloud vs. Git

Track changes	
Collaboration	
Versioning	
Syncing	
Access	
Data security	

Cloud services



- ✓ Documents
- ✓ Small data
- ✓ Presentations

Automated

Automated

Oftentimes only within organization / institution

Private / commercial

Git / GitHub / GitLab



- ✓ Code
- ✓ Data analytical projects

issue tracker, tracked contribution

Well-documented
(commit history)

Active / controlled
by user

Easily collaborate
across institutions

GitLab: on-premise
and custom
solutions