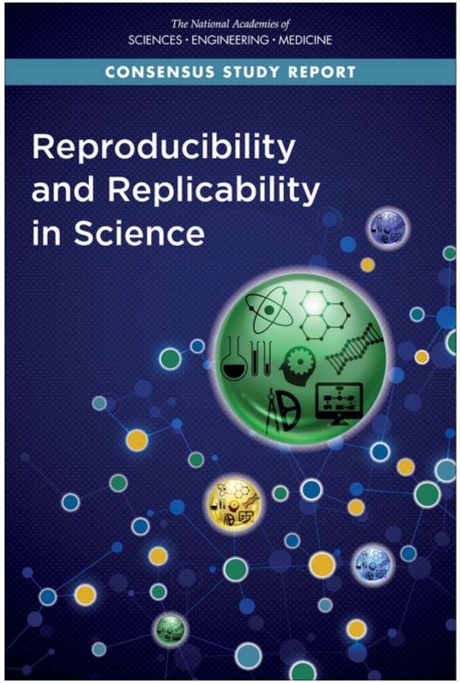
université PARIS-SACLAY | ÉCOLE DOCTORALE Structure et dynamique des systèmes vivants (SDSV)

Reproducibility at the service of Computational Biology

Thomas DENECKER

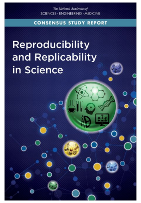
June 11, 2019

Reproducibility in Science in 2019


The National Academies of
SCIENCES · ENGINEERING · MEDICINE
CONSENSUS STUDY REPORT
Reproducibility and Replicability in Science

National Academies of Sciences, Engineering, and Medicine. 2019. Reproducibility and Replicability in Science. Washington, DC: The National Academies Press. https://doi.org/10.17226/25303.

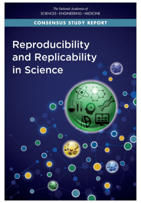
What is reproducibility?


The National Academies of SCIENCES · ENGINEERING · MEDICINE
CONSENSUS STUDY REPORT
Reproducibility and Replicability in Science

Reproducibility

Obtaining consistent computational results using the same input data, computational steps, methods, codes, and conditions of analysis

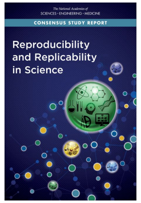
Reproducibility *versus* Replicability


The National Academies of
SCIENCES · ENGINEERING · MEDICINE
CONSENSUS STUDY REPORT
Reproducibility and Replicability in Science

Reproducibility

Obtaining consistent computational results using the same input data, computational steps, methods, codes, and conditions of analysis

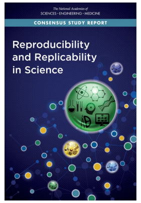
≠


The National Academies of
SCIENCES · ENGINEERING · MEDICINE
CONSENSUS STUDY REPORT
Reproducibility and Replicability in Science

Replicability

Obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data

Reproducibility *versus* Repeatability


CONSENSUS STUDY REPORT

Reproducibility and Replicability in Science

Reproducibility

Obtaining consistent computational results using the same input data, computational steps, methods, code, and conditions of analysis

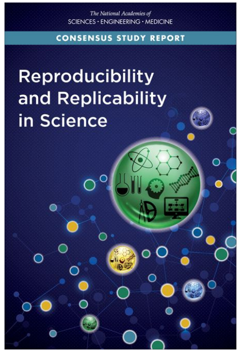
$\neq$

Repeatability (adapted from de Hans E. Plesser, *Front. Neuroinf.* , 2017)

frontiers in Neuroinformatics

Obtaining the closest possible results by performing an identical experiment (methods, equipment, experimenters, laboratory and conditions)

Recommendations for reproducibility in 2019


The National Academies of
SCIENCES · ENGINEERING · MEDICINE
CONSENSUS STUDY REPORT
Reproducibility and Replicability in Science

Description of the experimental part

Methods, instruments, procedures, measurements, experimental conditions

Description of the computational part

Steps in data analysis and technical choices

Description of the statistical part

Analytical decisions: when, how, why

Discussion of the choices and results obtained

In reality ?

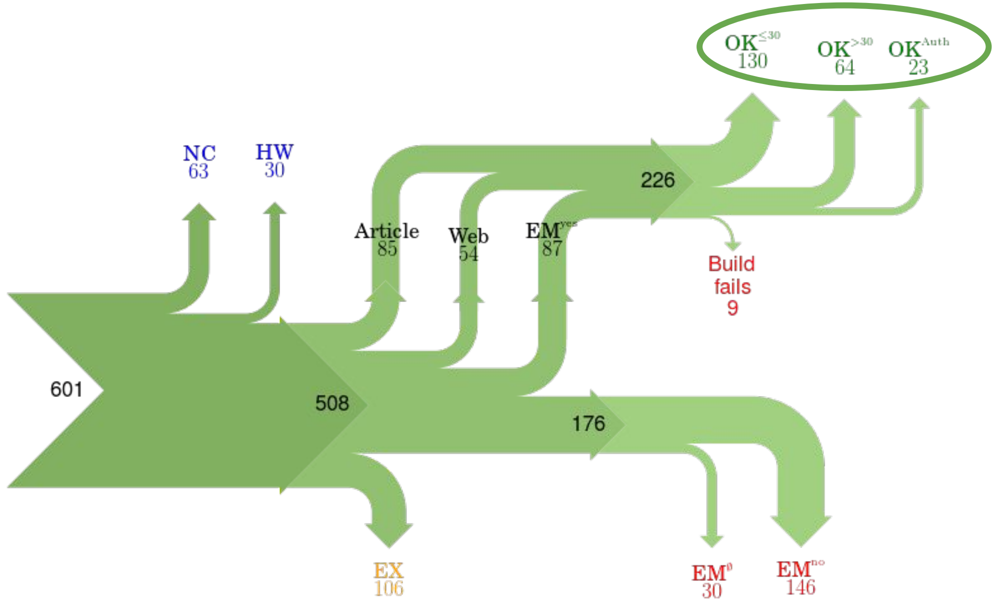
A problem of reproducibility in Biology

70 %

analyses in Experimental Biology **are not reproducible**

(Monya Baker, *Nature*, 2016)

In computer science



Collberg *et al*, University of Arizona TR 14-04, 2015

Bioinformatics is also affected...

Problems still too frequent

- **Impossibility to install tools**
  - OS not compatible
  - Dependency no longer available
- **Update of the tool rendering the codes unusable**
  - Python 2 and Python 3 !
  - Changing the arguments of the functions used (R)
- **Impossibility to reproduce the results of the computational analysis**
  - IDE : stable version of the language different according to the OS (Rstudio)
  - Package versions

Fortunately, there are solutions!

When bioinformatics meets development solutions

docker

CONDA

rmarkdown

git

SNAKEMAKE

jupyter

GitHub

...

An example of reproducibility in Bioinformatics

I2BC training with Claire Toffano-Nioche

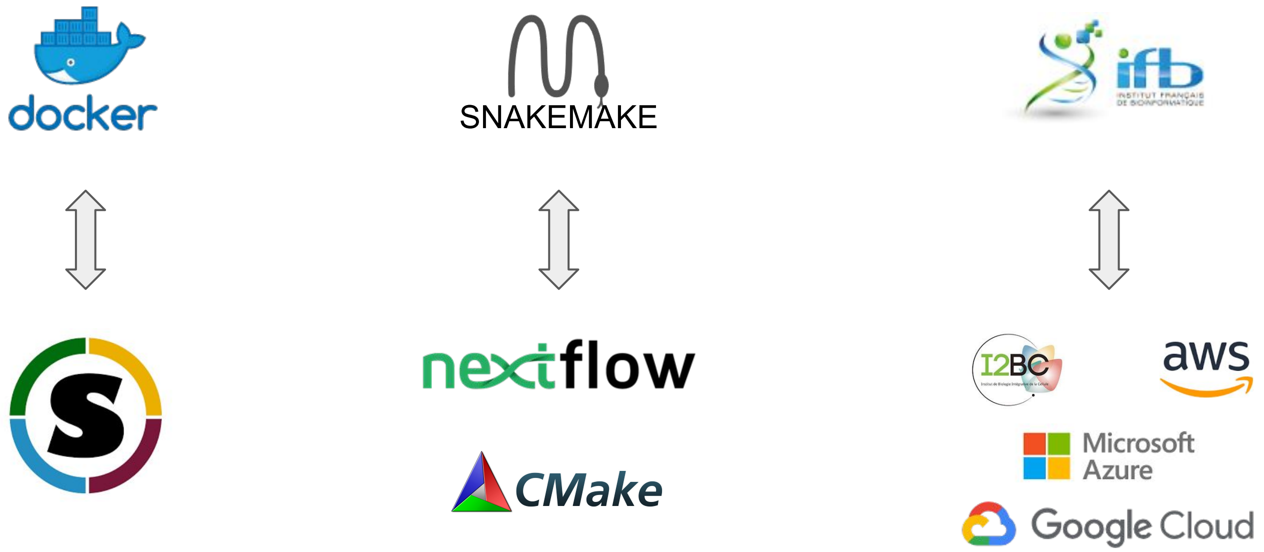
Use the FAIR data principles to make an analysis protocol reproducible and always obtain the same results from the same data

All courses and codes are open source

https://github.com/thomasdenecker/FAIR_Bioinfo

Proposal of a solution

Equivalents for each choice



...

The FAIR data principles

Findable **A**ccessible **I**nteroperable **R**eusable

How to use the FAIR data principles?

F

A

I

R

How to use the FAIR data principles?

Findable

- Tools used = references in their field
- Easy to find analysis protocol (Github pages)

A

I

R

How to use the FAIR data principles?

Findable

- Tools used = references in their field
- Easy to find analysis protocol (Github pages)

Accessible

- Available resources (Github, dockerhub)
- Open source tools (conda)

I

R

How to use the FAIR data principles?

Findable

- Tools used = references in their field
- Easy to find analysis protocol (Github pages)

Accessible

- Available resources (Github, dockerhub)
- Open source tools (conda)

Interoperable

- Cooperation of tools (snakemake, docker) both locally and on servers (cloud or cluster)

R

How to use the FAIR data principles?

Findable

- Tools used = references in their field
- Easy to find analysis protocol (Github pages)

Accessible

- Available resources (Github, dockerhub)
- Open source tools (conda)

Interoperable

- Cooperation of tools (snakemake, docker) both locally and on servers (cloud or cluster)

Reusable

- Protocol that can be simply replayed (snakemake) in the same way (Rmarkdown) in a virtual environment (docker)

Our creed : So FAIR !

FAIR raw data

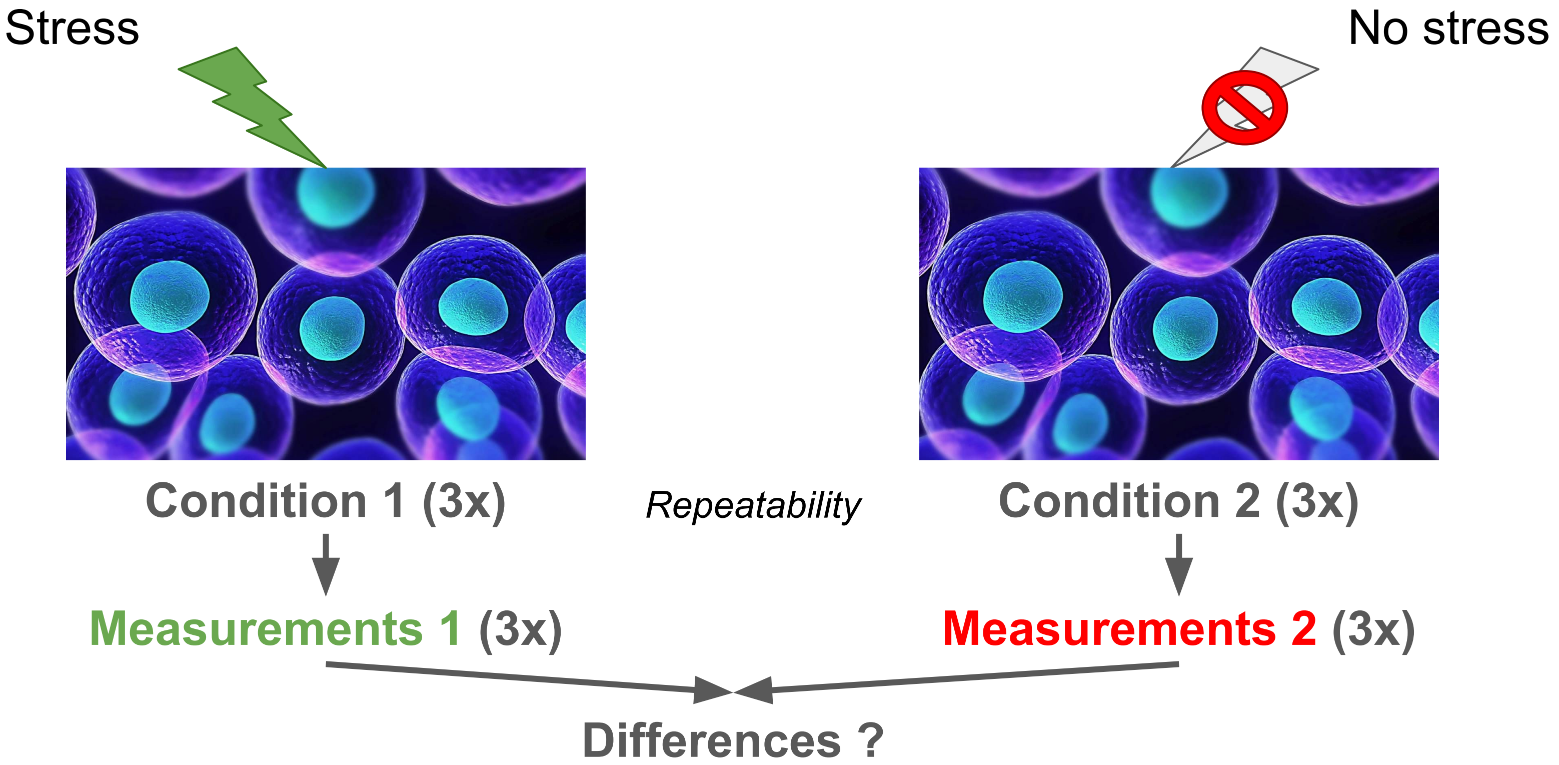
+

FAIR_bioinfo scripts/protocols

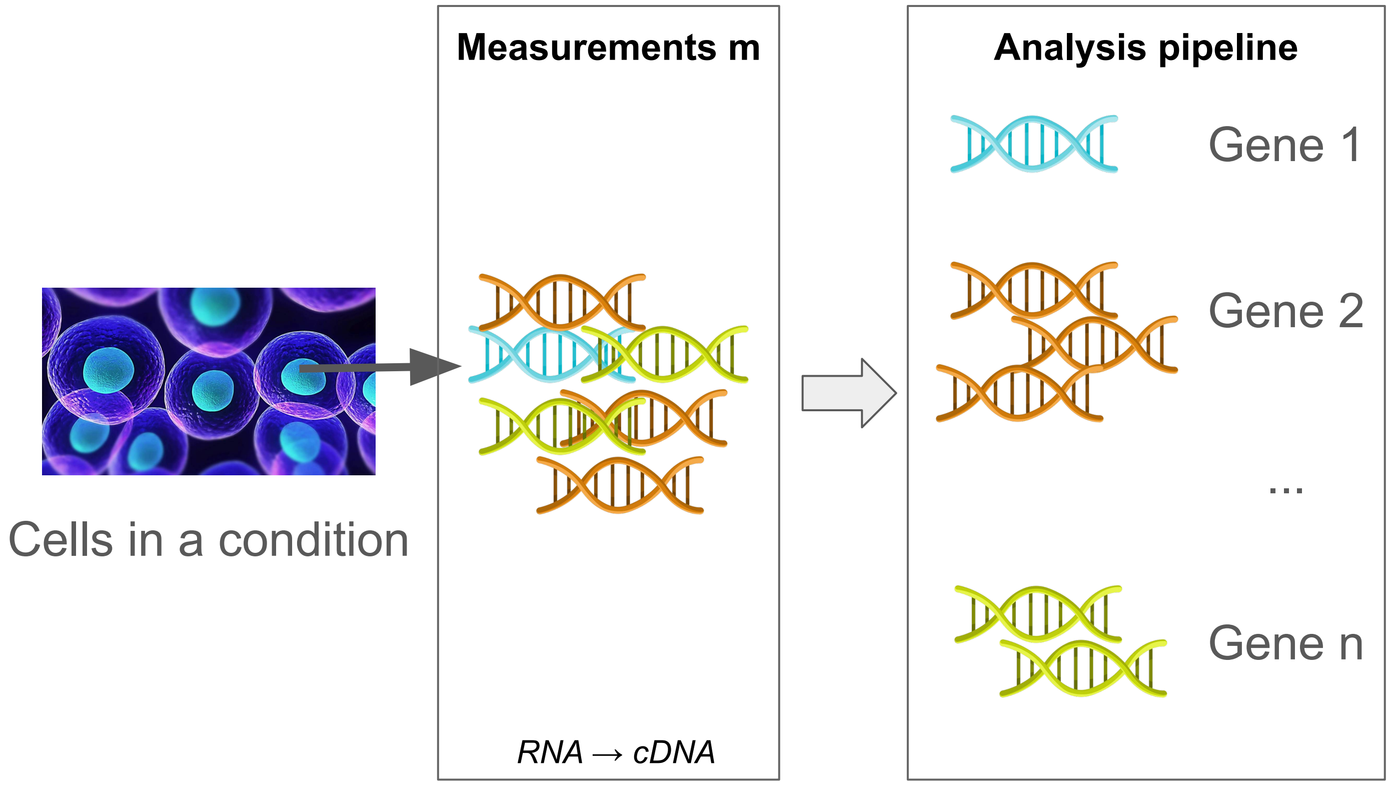
=

FAIR processed data

The data


Stress
No stress
Condition 1 (3x)
Repeatability
Condition 2 (3x)
Measurements 1 (3x)
Measurements 2 (3x)
Differences ?

Level of gene expression



Count table

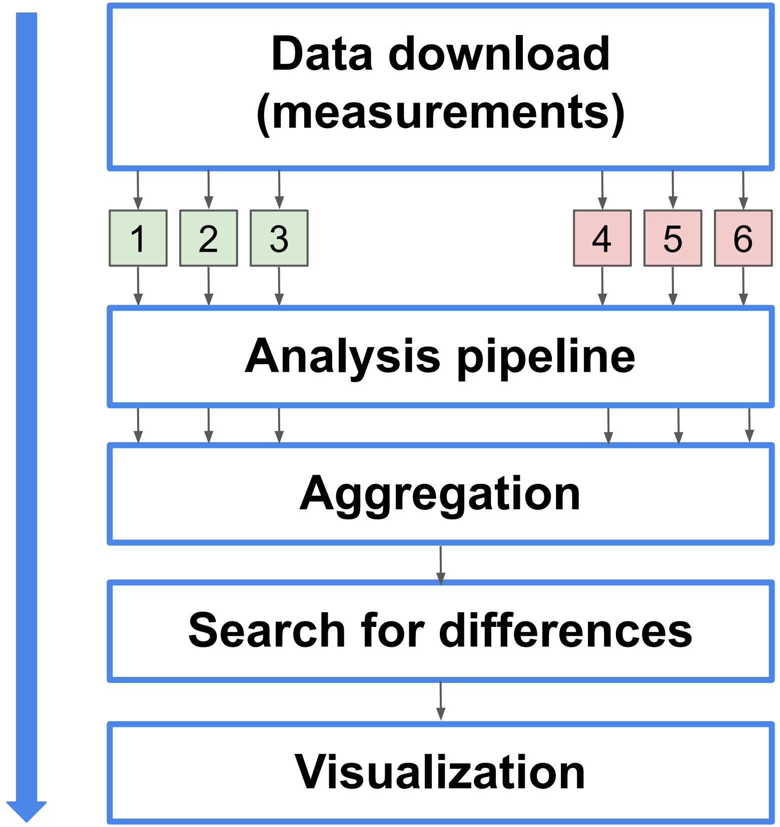
Gene 1	1
Gene 2	3
...	
Gene n	2

Differences between conditions

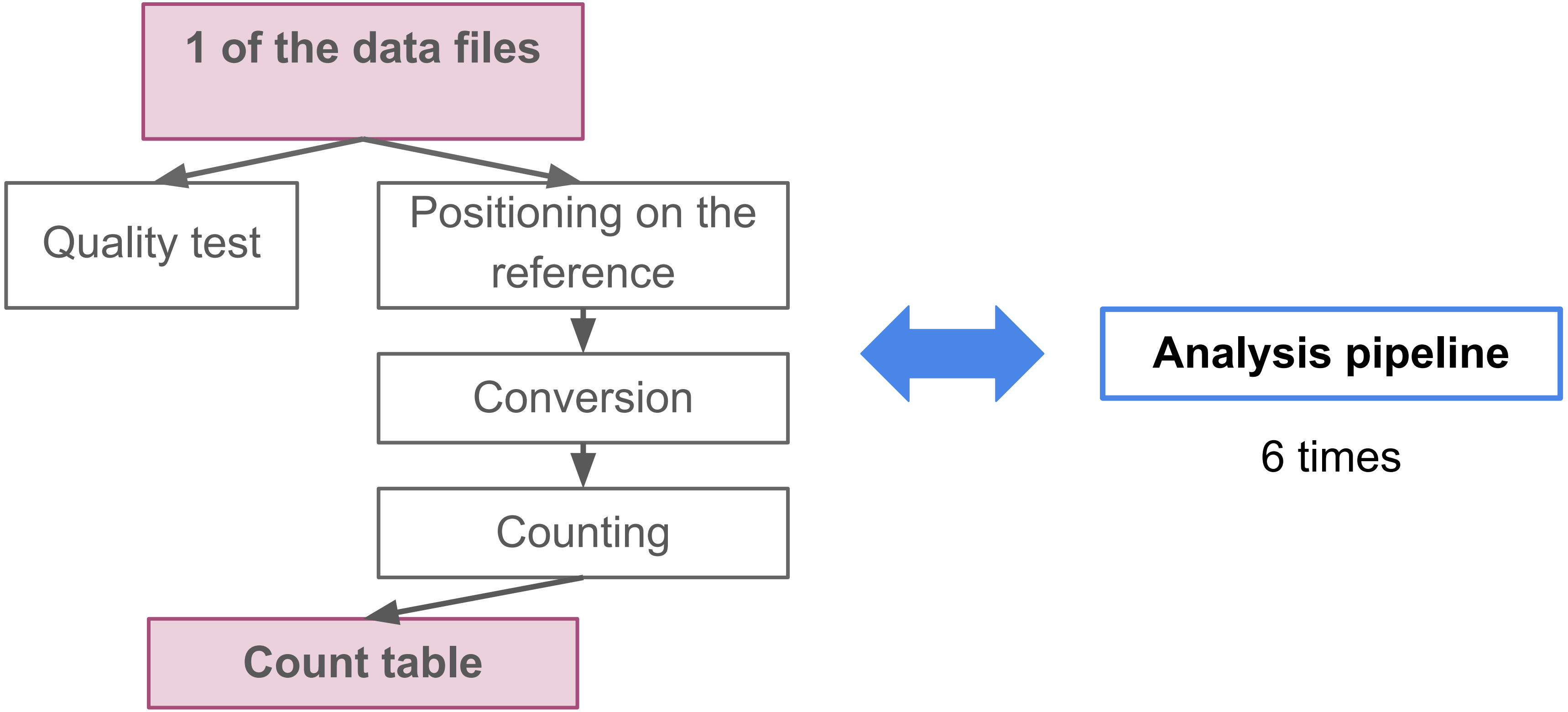
	Condition 1			**Condition 2**			
	1	**2**	**3**	**4**	**5**	**6**	
Gene 1	1	1	1	3	3	3	≠
Gene 2	3	3	3	1	1	1	≠
...	...	...	...	...	...	...	
Gene n	2	2	2	2	2	2	=

Conclusion
Genes 1 and 2 differentially expressed between condition 1 and 2

How to process this data?



Analysis pipeline



What is done mostly

1. Install the tools locally (sometimes writing an installation script)
2. Write a script to run all the analyses (not always...)
3. Sharing the script (by publishing, by email, USB key,...)

But low reproducibility ...

What are the solutions to be more reproducible in Bioinformatics?

A 7-step solution

Step 7 - Edit an analysis report

Step 6 - Portability of results exploration

Step 5 - Transfer the analysis to a server

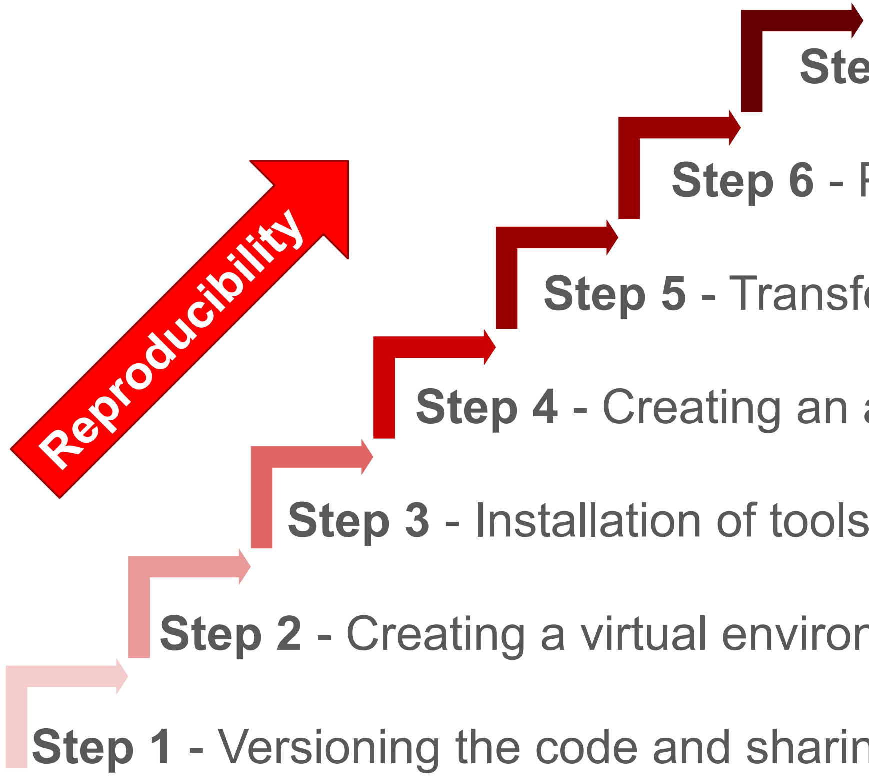
Step 4 - Creating an analysis script

Step 3 - Installation of tools by a manager

Step 2 - Creating a virtual environment

Step 1 - Versioning the code and sharing it

A 7-step solution



Step 7 - Edit an analysis report

Step 6 - Portability of results exploration

Step 5 - Transfer the analysis to a server

Step 4 - Creating an analysis script

Step 3 - Installation of tools by a manager

Step 2 - Creating a virtual environment

Step 1 - Versioning the code and sharing it

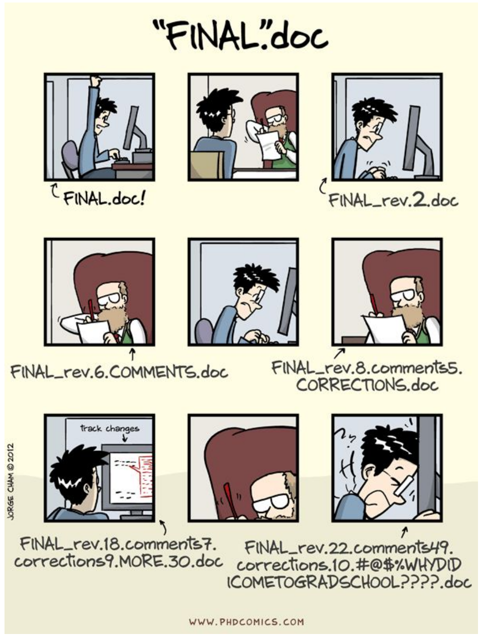
Step 1 - Versioning the code and sharing it

Why?

- Have the right version of the code
- Vision over time
- Open to the community

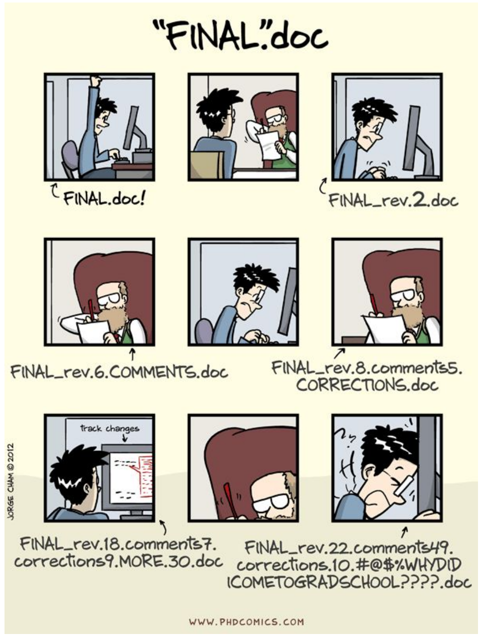
Step 1 - Versioning the code and sharing it

Before


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Step 1 - Versioning the code and sharing it

Before


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

git GitHub

Advantages

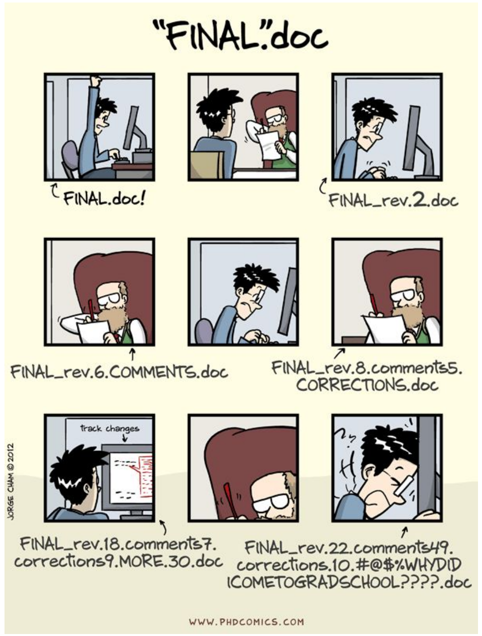
- Saving the code
- Simple to share
- Automatic version management

Disadvantages

- Not easy for novices

Step 1 - Versioning the code and sharing it

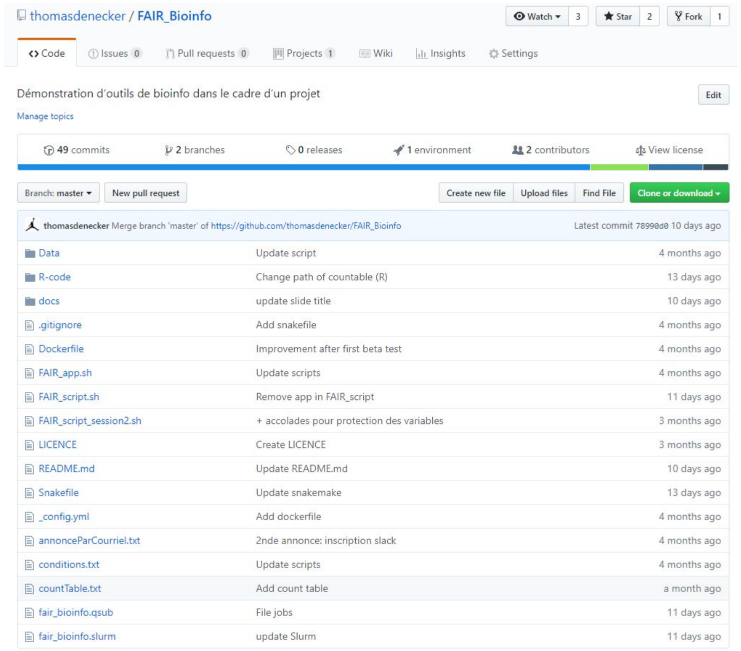
Before


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

git

GitHub

After



Démonstration d'outils de bioinfo dans le cadre d'un projet

thomasdenecker Merge branch 'master' of https://github.com/thomasdenecker/FAIR_Bioinfo — Latest commit 78990d0 10 days ago

Data	Update script	4 months ago
R-code	Change path of countable (R)	13 days ago
docs	update slide title	10 days ago
.gitignore	Add snakefile	4 months ago
Dockerfile	Improvement after first beta test	4 months ago
FAIR_app.sh	Update scripts	4 months ago
FAIR_script.sh	Remove app in FAIR_script	11 days ago
FAIR_script_session2.sh	+ accolades pour protection des variables	3 months ago
LICENCE	Create LICENCE	3 months ago
README.md	Update README.md	10 days ago
Snakefile	Update snakemake	13 days ago
_config.yml	Add dockerfile	4 months ago
annonceParCourriel.txt	2nde annonce: inscription slack	4 months ago
conditions.txt	Update scripts	4 months ago
countTable.txt	Add count table	a month ago
fair_bioinfo.qsub	File jobs	11 days ago
fair_bioinfo.slurm	update Slurm	11 days ago

Step 2 - Creating a virtual environment

Why ?

- Same the environment for all
- Sharing the environment

Step 2 - Creating a virtual environment

Before

Step 2 - Creating a virtual environment

Before

Advantages

- Fast and lightweight
- Portable
- Easy to share and deploy

Disadvantages

- root
- Up-to-date system

Step 2 - Creating a virtual environment

Before

After : Ubuntu 16.04

```
$ cat > dockerfile
FROM ubuntu:16.04
RUN apt-get update

# Set environment variables
ENV HOME /root

# Define working directory
WORKDIR /root

# Define default command
CMD ["bash"]

$ docker --tag=toto build .
$ docker run toto
```

Step 3 - Installation of the tools by a manager

Why ?

- Good version
- Simply install

Step 3 - Installation of the tools by a manager

Before : FastQC

1) Download the source
2) Unzip the file
3) Install and update Java (many problems)
4) Changing rights

Step 3 - Installation of the tools by a manager

Before : FastQC

1) Download the source
2) Unzip the file
3) Install and update Java (many problems)
4) Changing rights

→

CONDA

Advantages

- Easy to install manager
- Easy package installation
- Version management

Disadvantages

- May be heavy (miniconda solution)
- Missing packages (R)

Step 3 - Installation of the tools by a manager

Before : FastQC

1) Download the source
2) Unzip the file
3) Install and update Java (many problems)
4) Changing rights

CONDA

Après

```
$ conda install -c bioconda -y
fastqc=0.11.2
```

All the tools used in the protocol are available on Conda (https://anaconda.org/) : bowtie2, samtools, htseqcount, aspera, snakemake, ...

Installation as easy as that

Step 4 - Creating an analysis script

Why ?

- Have a reproducible analysis script
- Do not repeat what has already been done
- Parallelize

Step 4 - Creating an analysis script

Before (Shell script)

```
for sample in `ls *.fastq.gz`
do
  fastqc ${sample}
done
```

Step 4 - Creating an analysis script

Before (Shell script)

```
for sample in `ls *.fastq.gz`
do
  fastqc ${sample}
done
```

→

SNAKEMAKE

Advantages

- Workflow (job management)
- Powerful and fast

Disadvantages

- A logic to be taken
- Syntax less simple than shell script

Step 4 - Creating an analysis script

Before (Shell script)

```
for sample in `ls *.fastq.gz`
do
  fastqc ${sample}
done
```

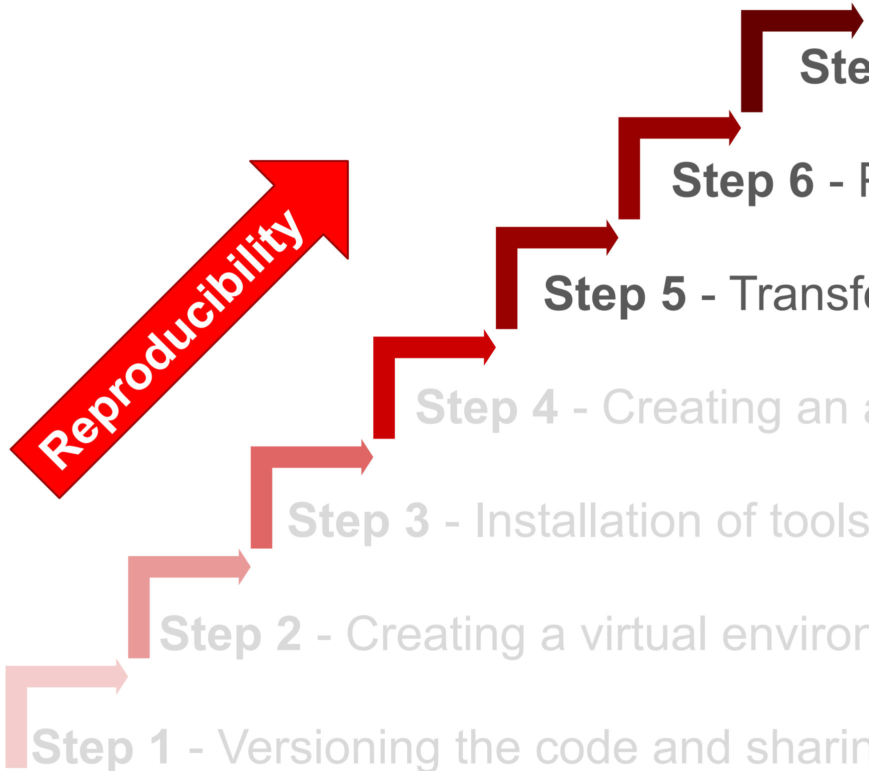
After (Snakefile)

```
$ cat > Snakefile
SAMPLES, =
glob_wildcards("./samples/{smp}.fastq.gz")

rule final:
input:expand("fastqc/{smp}/{smp}_fastqc.zip"
,smp=SAMPLES)

rule fastqc:
   input:  "samples/{smp}.fastq.gz"
   output:"fastqc/{smp}/{smp}_fastqc.zip"
   message: """Quality check"""
   shell: """fastqc {input} --outdir
fastqc/{wildcards.smp}"""
$ snakemake
```

Shorter to write but not to execute

Snakemake = Parallel

A 7-step solution



Step 7 - Edit an analysis report

Step 6 - Portability of results exploration

Step 5 - Transfer the analysis to a server

Step 4 - Creating an analysis script

Step 3 - Installation of tools by a manager

Step 2 - Creating a virtual environment

Step 1 - Versioning the code and sharing it

Step 5 - Transfer the analysis to a server

Why ?

- Controlled environment
- Offset of the analysis

Step 5 - Transfer the analysis to a server

Before

Adaptation locally and on servers that is difficult or even unmanaged ...

Step 5 - Transfer the analysis to a server

Before

Adaptation locally and on servers that is difficult or even unmanaged ...

Advantages

- Easy to set up
- Increase in power (cloud or cluster)
- For everyone

Disadvantages

- Not easy for novices

Step 5 - Transfer the analysis to a server

Before

Adaptation locally and on servers that is difficult or even unmanaged ...

After

```
$ git clone
https://github.com/thomasdenecker/FAIR_Bioinfo

$ cd FAIR_Bioinfo

$ sudo docker run --rm -d -p 80:8888 --name
fair_bioinfo -v ${PWD}:/home/rstudio
tdenecker/fair_bioinfo bash ./FAIR_script.sh
```

The protocol is running !

Step 6 - Portability of the results exploration

Why ?

- Make it easy to explore
- Easy to share

Step 6 - Portability of the results exploration

Before : R terminal

```
dds <- DESeqDataSetFromMatrix(countData =
cts,colData = coldata, design= ~ batch +
condition)

dds <- DESeq(dds)
resultsNames(dds) # lists the
coefficients
res <- results(dds, name =
"condition_trt_vs_untrt")

# or to shrink log fold changes
# association with condition:
res <- lfcShrink(dds,
coef="condition_trt_vs_untrt",
type="apeglm")
```

Step 6 - Portability of the results exploration

Before : R terminal

```
dds <- DESeqDataSetFromMatrix(countData =
cts,colData = coldata, design= ~ batch +
condition)

dds <- DESeq(dds)
resultsNames(dds) # lists the
coefficients
res <- results(dds, name =
"condition_trt_vs_untrt")

# or to shrink log fold changes
# association with condition:
res <- lfcShrink(dds,
coef="condition_trt_vs_untrt",
type="apeglm")
```

Advantages

- Portable (HTML)
- Accessible everywhere
- Interactive (configurable, dynamic graphs,...)

Disadvantages

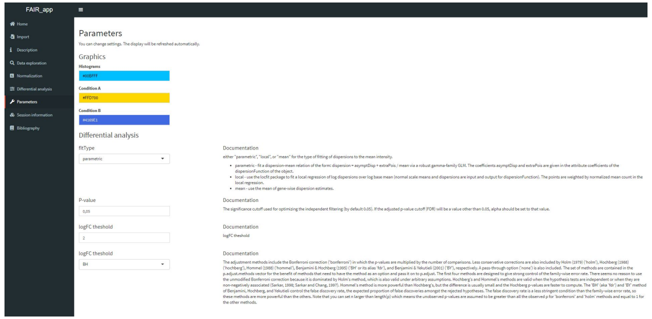
- Mixing R and HTML

Step 6 - Portability of the results exploration

Before : R terminal

```
dds <- DESeqDataSetFromMatrix(countData =
cts,colData = coldata, design= ~ batch +
condition)

dds <- DESeq(dds)
resultsNames(dds) # lists the
coefficients
res <- results(dds, name =
"condition_trt_vs_untrt")

# or to shrink log fold changes
# association with condition:
res <- lfcShrink(dds,
coef="condition_trt_vs_untrt",
type="apeglm")
```

After



Parameters

You can change settings. The display will be refreshed automatically.

Graphics

Histograms: #00BFFF

Condition A: #FFD700

Condition B: #4169E1

Differential analysis

fitType: parametric

Documentation

either "parametric", "local", or "mean" for the type of fitting of dispersions to the mean intensity.

- parametric - fit a dispersion-mean relation of the form: dispersion = asymptDisp + extraPois / mean via a robust gamma-family GLM. The coefficients asymptDisp and extraPois are given in the attribute coefficients of the dispersionFunction of the object.
- local - use the locfit package to fit a local regression of log dispersions over log base mean (normal scale means and dispersions are input and output for dispersionFunction). The points are weighted by normalized mean count in the local regression.
- mean - use the mean of gene-wise dispersion estimates.

P-value: 0,05

Documentation

The significance cutoff used for optimizing the independent filtering (by default 0.05). If the adjusted p-value cutoff (FDR) will be a value other than 0.05, alpha should be set to that value.

logFC theshold: 2

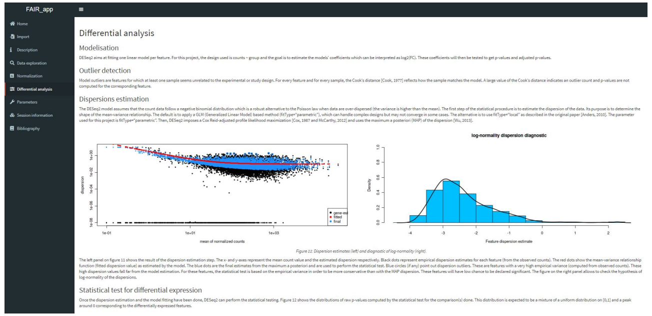
Documentation

logFC theshold

logFC theshold: BH

Documentation

The adjustment methods include the Bonferroni correction ('bonferroni') in which the p-values are multiplied by the number of comparisons. Less conservative corrections are also included by Holm (1979) ('holm'), Hochberg (1988) ('hochberg'), Hommel (1988) ('hommel'), Benjamini & Hochberg (1995) ('BH' or its alias 'fdr'), and Benjamini & Yekutieli (2001) ('BY'), respectively. A pass-through option ('none') is also included. The set of methods are contained in the p.adjust.methods vector for the benefit of methods that need to have the method as an option and pass it on to p.adjust. The first four methods are designed to give strong control of the family-wise error rate. There seems no reason to use the unmodified Bonferroni correction because it is dominated by Holm's method, which is also valid under arbitrary assumptions. Hochberg's and Hommel's methods are valid when the hypothesis tests are independent or when they are non-negatively associated (Sarkar, 1998; Sarkar and Chang, 1997). Hommel's method is more powerful than Hochberg's, but the difference is usually small and the Hochberg p-values are faster to compute. The 'BH' (aka 'fdr') and 'BY' method of Benjamini, Hochberg, and Yekutieli control the false discovery rate, the expected proportion of false discoveries amongst the rejected hypotheses. The false discovery rate is a less stringent condition than the family-wise error rate, so these methods are more powerful than the others. Note that you can set n larger than length(p) which means the unobserved p-values are assumed to be greater than all the observed p for 'bonferroni' and 'holm' methods and equal to 1 for the other methods.



Differential analysis

Modelisation

DESeq2 aims at fitting one linear model per feature. For this project, the design used is counts ~ group and the goal is to estimate the models' coefficients which can be interpreted as log2(FC). These coefficients will then be tested to get p-values and adjusted p-values.

Outlier detection

Model outliers are features for which at least one sample seems unrelated to the experimental or study design. For every feature and for every sample, the Cook's distance [Cook, 1977] reflects how the sample matches the model. A large value of the Cook's distance indicates an outlier count and p-values are not computed for the corresponding feature.

Dispersions estimation

The DESeq2 model assumes that the count data follow a negative binomial distribution which is a robust alternative to the Poisson law when data are over-dispersed (the variance is higher than the mean). The first step of the statistical procedure is to estimate the dispersion of the data. Its purpose is to determine the shape of the mean-variance relationship. The default is to apply a GLM (Generalized Linear Model) based method (fitType="parametric"), which can handle complex designs but may not converge in some cases. The alternative is to use fitType="local" as described in the original paper [Anders, 2010]. The parameter used for this project is fitType="parametric". Then, DESeq2 imposes a Cox Reid-adjusted profile likelihood maximization [Cox, 1987 and McCarthy, 2012] and uses the maximum a posteriori (MAP) of the dispersion [Wu, 2013].

Figure 11: Dispersion estimates (left) and diagnostic of log-normality (right).

The left panel on figure 11 shows the result of the dispersion estimation step. The x- and y-axes represent the mean count value and the estimated dispersion respectively. Black dots represent empirical dispersion estimates for each feature (from the observed counts). The red dots show the mean-variance relationship function (fitted dispersion value) as estimated by the model. The blue dots are the final estimates from the maximum a posteriori and are used to perform the statistical test. Blue circles (if any) point out dispersion outliers. These are features with a very high empirical variance (computed from observed counts). These high dispersion values fall far from the model estimation. For these features, the statistical test is based on the empirical variance in order to be more conservative than with the MAP dispersion. These features will have low chance to be declared significant. The figure on the right panel allows to check the hypothesis of log-normality of the dispersions.

Statistical test for differential expression

Once the dispersion estimation and the model fitting have been done, DESeq2 can perform the statistical testing. Figure 12 shows the distributions of raw p-values computed by the statistical test for the comparison(s) done. This distribution is expected to be a mixture of a uniform distribution on [0,1] and a peak around 0 corresponding to the differentially expressed features.

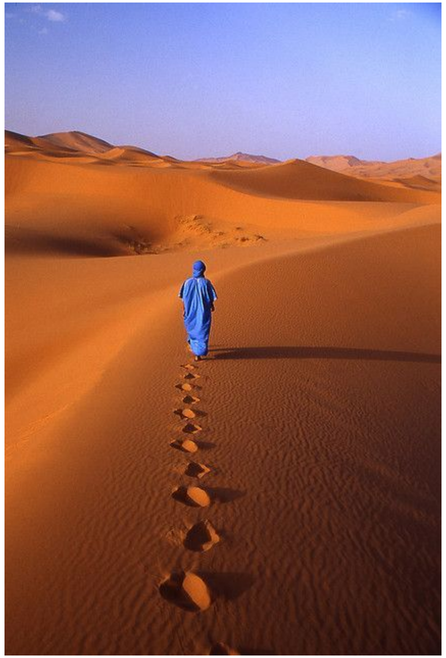
Step 7 - Edit an analysis report

Why ?

- Have a trace of the analysis (date, time, parameters,...)
- Store tool versions

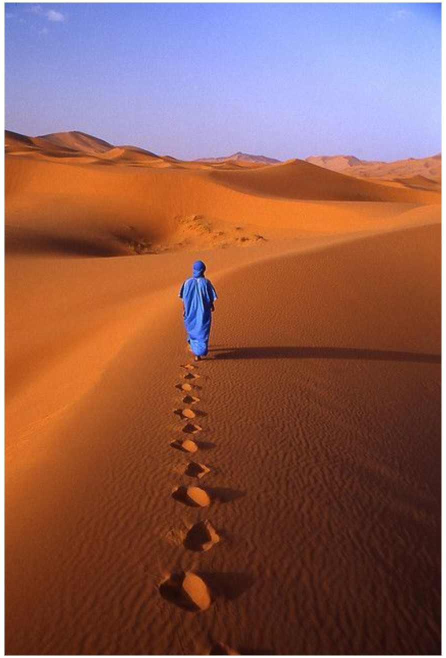
Step 7 - Edit an analysis report

Before



Step 7 - Edit an analysis report

Before



→

rmarkdown jupyter

Advantages

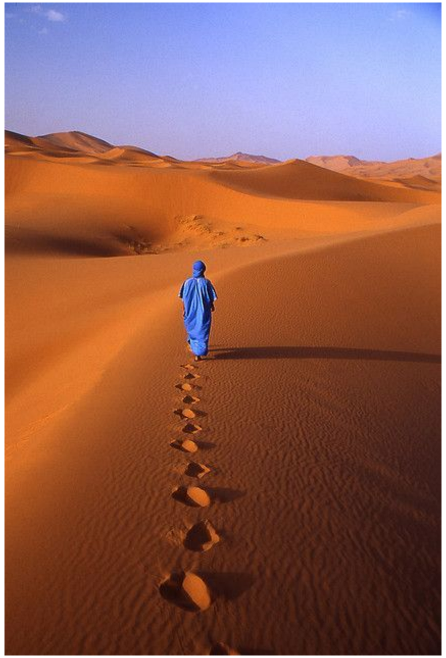
- Simple syntax (Markdown)
- Sharing (PDF, HTML,...)

Disadvantages

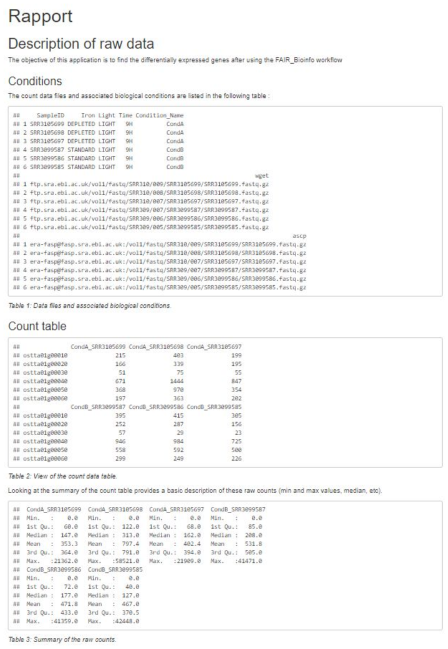
- Rare visualization problems in LaTeX

Step 7 - Edit an analysis report

Before



After



Rapport

Description of raw data

The objective of this application is to find the differentially expressed genes after using the FAIR_Bioinfo workflow

Conditions

The count data files and associated biological conditions are listed in the following table :

```
##     SampleID     Iron Light Time Condition_Name
## 1 SRR3105699 DEPLETED LIGHT   9H          CondA
## 2 SRR3105698 DEPLETED LIGHT   9H          CondA
## 3 SRR3105697 DEPLETED LIGHT   9H          CondA
## 4 SRR3099587 STANDARD LIGHT   9H          CondB
## 5 SRR3099586 STANDARD LIGHT   9H          CondB
## 6 SRR3099585 STANDARD LIGHT   9H          CondB
##                                                                   wget
## 1 ftp.sra.ebi.ac.uk/vol1/fastq/SRR310/009/SRR3105699/SRR3105699.fastq.gz
## 2 ftp.sra.ebi.ac.uk/vol1/fastq/SRR310/008/SRR3105698/SRR3105698.fastq.gz
## 3 ftp.sra.ebi.ac.uk/vol1/fastq/SRR310/007/SRR3105697/SRR3105697.fastq.gz
## 4 ftp.sra.ebi.ac.uk/vol1/fastq/SRR309/007/SRR3099587/SRR3099587.fastq.gz
## 5 ftp.sra.ebi.ac.uk/vol1/fastq/SRR309/006/SRR3099586/SRR3099586.fastq.gz
## 6 ftp.sra.ebi.ac.uk/vol1/fastq/SRR309/005/SRR3099585/SRR3099585.fastq.gz
##                                                                           ascp
## 1 era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR310/009/SRR3105699/SRR3105699.fastq.gz
## 2 era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR310/008/SRR3105698/SRR3105698.fastq.gz
## 3 era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR310/007/SRR3105697/SRR3105697.fastq.gz
## 4 era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR309/007/SRR3099587/SRR3099587.fastq.gz
## 5 era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR309/006/SRR3099586/SRR3099586.fastq.gz
## 6 era-fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR309/005/SRR3099585/SRR3099585.fastq.gz
```

Table 1: Data files and associated biological conditions.

Count table

```
##                CondA_SRR3105699 CondA_SRR3105698 CondA_SRR3105697
## ostta01g00010               215              403              199
## ostta01g00020               166              339              195
## ostta01g00030                51               75               55
## ostta01g00040               671             1444              847
## ostta01g00050               368              970              354
## ostta01g00060               197              363              202
##                CondB_SRR3099587 CondB_SRR3099586 CondB_SRR3099585
## ostta01g00010               395              415              305
## ostta01g00020               252              287              156
## ostta01g00030                57               29               23
## ostta01g00040               946              984              725
## ostta01g00050               558              592              500
## ostta01g00060               299              249              226
```

Table 2: View of the count data table.

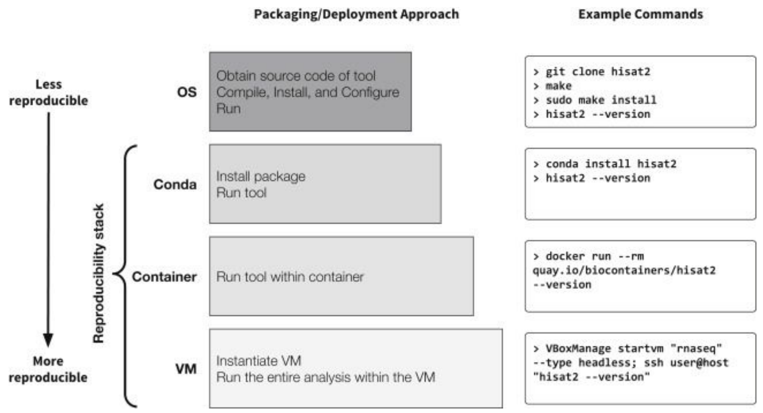
Looking at the summary of the count table provides a basic description of these raw counts (min and max values, median, etc).

```
##  CondA_SRR3105699  CondA_SRR3105698  CondA_SRR3105697  CondB_SRR3099587
##  Min.   :    0.0   Min.   :    0.0   Min.   :    0.0   Min.   :    0.0
##  1st Qu.:   60.0   1st Qu.:  122.0   1st Qu.:   68.0   1st Qu.:   85.0
##  Median :  147.0   Median :  313.0   Median :  162.0   Median :  208.0
##  Mean   :  353.3   Mean   :  797.4   Mean   :  402.4   Mean   :  531.8
##  3rd Qu.:  364.0   3rd Qu.:  791.0   3rd Qu.:  394.0   3rd Qu.:  505.0
##  Max.   :21362.0   Max.   :58521.0   Max.   :21909.0   Max.   :41471.0
##  CondB_SRR3099586  CondB_SRR3099585
##  Min.   :    0.0   Min.   :    0.0
##  1st Qu.:   72.0   1st Qu.:   40.0
##  Median :  177.0   Median :  127.0
##  Mean   :  471.8   Mean   :  467.0
##  3rd Qu.:  433.0   3rd Qu.:  370.5
##  Max.   :41359.0   Max.   :42448.0
```

Table 3: Summary of the raw counts.

Conclusion

What is our level of reproducibility?


Packaging/Deployment Approach
Example Commands
Less reproducible
More reproducible
Reproducibility stack
OS
Obtain source code of tool
Compile, Install, and Configure
Run
Conda
Install package
Run tool
Container
Run tool within container
VM
Instantiate VM
Run the entire analysis within the VM

```
> git clone hisat2
> make
> sudo make install
> hisat2 --version
```

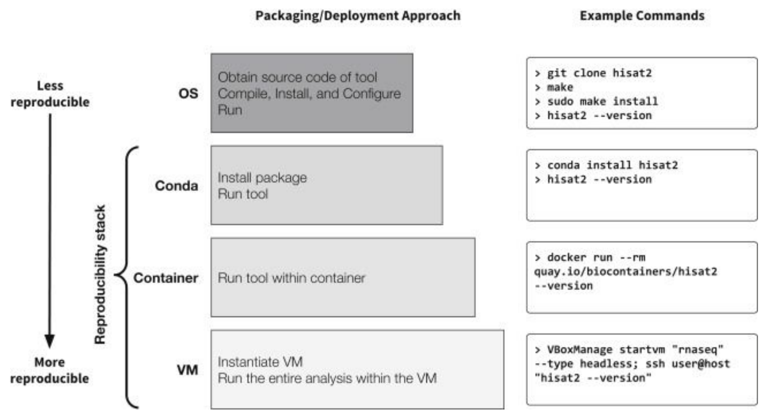
```
> conda install hisat2
> hisat2 --version
```

```
> docker run --rm
quay.io/biocontainers/hisat2
--version
```

```
> VBoxManage startvm "rnaseq"
--type headless; ssh user@host
"hisat2 --version"
```

Practical Computational Reproducibility in the Life Sciences, Björn Grüning *et al*, 2018

What is our level of reproducibility?


Packaging/Deployment Approach
Example Commands
Less reproducible
More reproducible
Reproducibility stack
OS
Conda
Container
VM
Obtain source code of tool
Compile, Install, and Configure
Run
Install package
Run tool
Run tool within container
Instantiate VM
Run the entire analysis within the VM

```
> git clone hisat2
> make
> sudo make install
> hisat2 --version
```

```
> conda install hisat2
> hisat2 --version
```

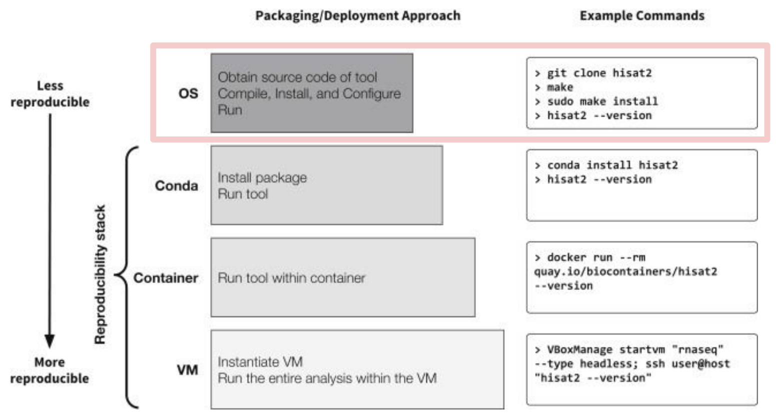
```
> docker run --rm
quay.io/biocontainers/hisat2
--version
```

```
> VBoxManage startvm "rnaseq"
--type headless; ssh user@host
"hisat2 --version"
```

Practical Computational Reproducibility in the Life Sciences, Björn Grüning *et al*, 2018

FAIR bioinfo

What is our level of reproducibility?



Packaging/Deployment Approach
Example Commands
Less reproducible
More reproducible
Reproducibility stack
OS
Obtain source code of tool
Compile, Install, and Configure
Run
Conda
Install package
Run tool
Container
Run tool within container
VM
Instantiate VM
Run the entire analysis within the VM

```
> git clone hisat2
> make
> sudo make install
> hisat2 --version
```

```
> conda install hisat2
> hisat2 --version
```

```
> docker run --rm
quay.io/biocontainers/hisat2
--version
```

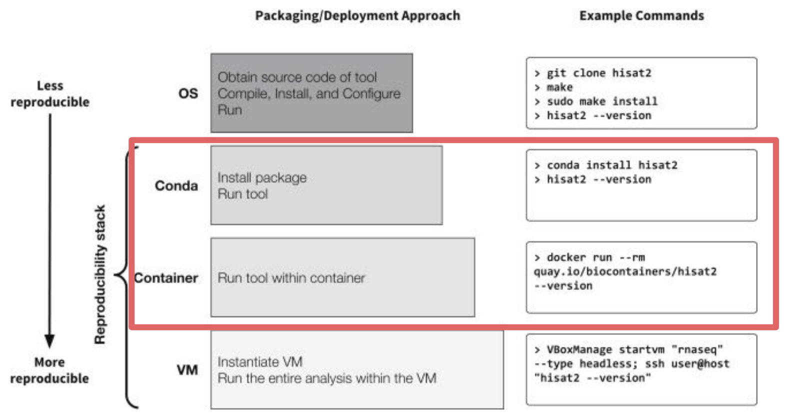
```
> VBoxManage startvm "rnaseq"
--type headless; ssh user@host
"hisat2 --version"
```

Practical Computational Reproducibility in the Life Sciences, Björn Grüning *et al*, 2018

1 git GitHub

FAIR bioinfo

What is our level of reproducibility?


Packaging/Deployment Approach
Example Commands
Less reproducible
More reproducible
Reproducibility stack
OS
Conda
Container
VM

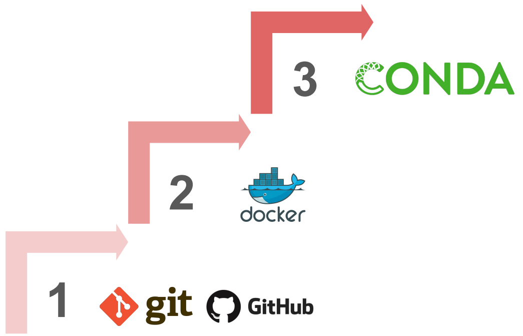
	Packaging/Deployment Approach	Example Commands

OS	Obtain source code of tool
	Compile, Install, and Configure
	Run
		`> git clone hisat2`
		`> make`
		`> sudo make install`
		`> hisat2 --version`

Conda	Install package
	Run tool
		`> conda install hisat2`
		`> hisat2 --version`

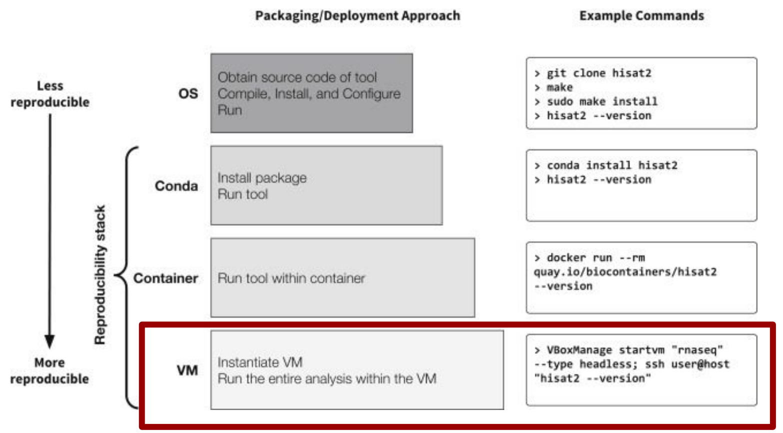
Container	Run tool within container	`> docker run --rm quay.io/biocontainers/hisat2 --version`
VM	Instantiate VM Run the entire analysis within the VM	`> VBoxManage startvm "rnaseq" --type headless; ssh user@host "hisat2 --version"`

Practical Computational Reproducibility in the Life Sciences, Björn Grüning *et al*, 2018



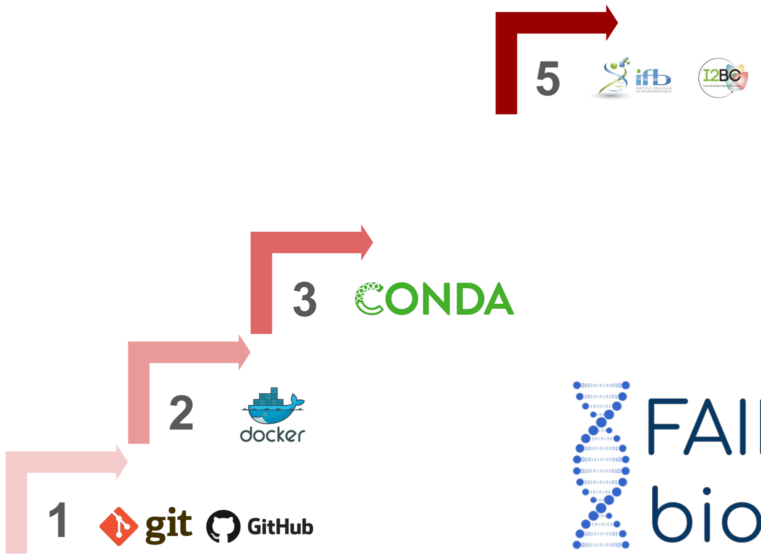
FAIR bioinfo

What is our level of reproducibility?



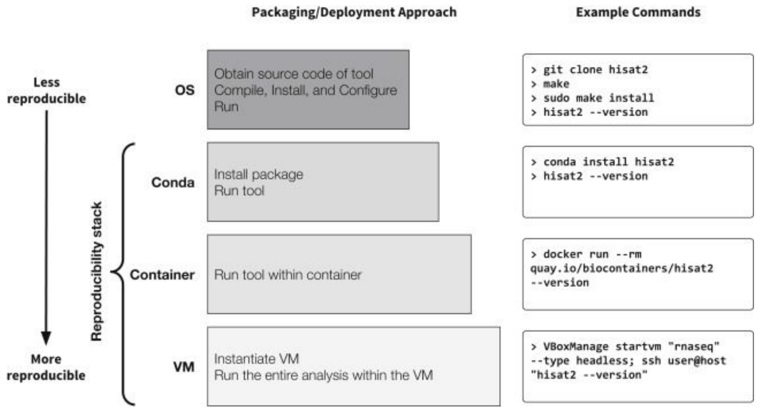
		Packaging/Deployment Approach	Example Commands
Less reproducible	OS	Obtain source code of tool Compile, Install, and Configure Run	> git clone hisat2 > make > sudo make install > hisat2 --version
	Conda	Install package Run tool	> conda install hisat2 > hisat2 --version
	Container	Run tool within container	> docker run --rm quay.io/biocontainers/hisat2 --version
More reproducible	VM	Instantiate VM Run the entire analysis within the VM	> VBoxManage startvm "rnaseq" --type headless; ssh user@host "hisat2 --version"

Practical Computational Reproducibility in the Life Sciences, Björn Grüning *et al*, 2018



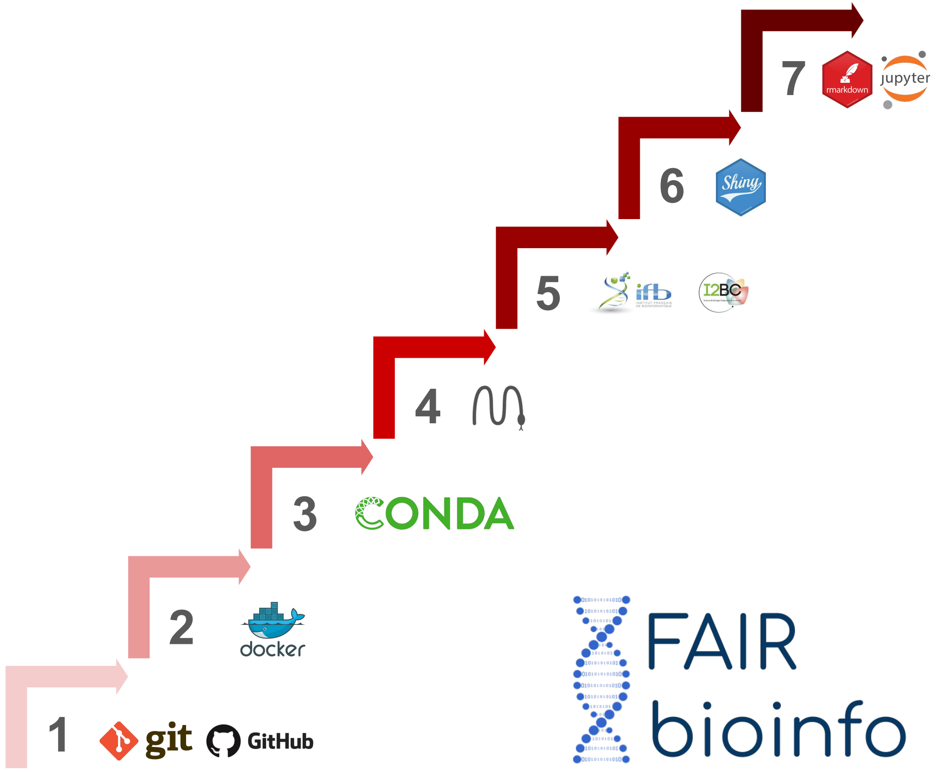
FAIR bioinfo

What is our level of reproducibility?


Less reproducible
More reproducible
Reproducibility stack

	Packaging/Deployment Approach	Example Commands
OS	Obtain source code of tool Compile, Install, and Configure Run	> git clone hisat2 > make > sudo make install > hisat2 --version
Conda	Install package Run tool	> conda install hisat2 > hisat2 --version
Container	Run tool within container	> docker run --rm quay.io/biocontainers/hisat2 --version
VM	Instantiate VM Run the entire analysis within the VM	> VBoxManage startvm "rnaseq" --type headless; ssh user@host "hisat2 --version"

Practical Computational Reproducibility in the Life Sciences, Björn Grüning *et al*, 2018



FAIR bioinfo

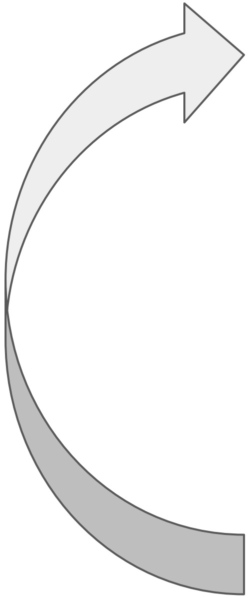
Take home messages

A real reflection on the reproducibility of analyses in Bioinformatics

Proposal of a solution that helps to make any analytical protocol reproducible

Reproducibility is an real plus value for Bioinformatics!

A virtuous circle



FAIR raw data

+

FAIR_bioinfo scripts/protocols

=

FAIR processed data