

# FAIR\_bioinfo : Open Science and FAIR principles in a bioinformatics project

How to make a bioinformatics project more reproducible

C. Hernandez<sup>1</sup>    T. Denecker<sup>2</sup>    J. Seiler<sup>2</sup>    G. Le Corguillé<sup>2</sup>  
C. Toffano-Nioche<sup>1</sup>

<sup>1</sup>Institute for Integrative Biology of the Cell (I2BC)  
UMR 9198, Université Paris-Sud, CNRS, CEA  
91190 - Gif-sur-Yvette, France

<sup>2</sup>IFB Core Cluster taskforce

June 2021

# General information

## Practical information:

- Dates: June 28th - 30th
- Location: Institut des Systèmes Complexes, 113 rue Nationale, 75013-Paris
- Courses: 9:00 to 17:30
- Meal: 12:30-14:00
- Pauses: 10:30-11:00 + 15:30-16:00
- 2 days of courses + 1 day of course building

## Round table:

- Teachers
- Learners

## Ressources:



- GitLab
- L<sup>A</sup>T<sub>E</sub>X

# Training schedule

## Day 1:

- Introduction to reproducibility
- History management (3 Practical Sessions, git, GitHub)
- Control your development environment (1 PS, CONDA)
- Encapsulation (2 PS, docker)

## Day 2:

- Workflow (2 PS, SNAKEMAKE)
- Traceability with notebooks (2 PS, jupyter, zenodo)
- IFB resources (2 PS, slurm, singularity)
- Sharing and disseminating ( GitHub, zenodo)
- Conclusion

## Day 3:

- Empowerment and improvement of resources

# Table of contents

- 1 Introduction to reproducibility
- 2 History management
- 3 Control your development environment
- 4 Workflow
- 5 Tracability with Notebook
  - Introduction
  - Markdown
- 6 IFB resources
- 7 Sharing and dissemination
- 8 Conclusion
- 9 3rd Day

# Literate programming

# Introduction

What is literate programming ?

Let us change our traditional attitude to the construction of programs:  
Instead of imagining that our main task is to instruct a computer what to do,  
let us concentrate rather on explaining to humans what we want the  
computer to do.

— Donald E. Knuth, Literate Programming, 1984

# Introduction

What is literate programming ?

## Definition

"Literate programming is a programming paradigm introduced by Donald Knuth in which a computer program is given an explanation of its logic in a natural language, such as English, interspersed with snippets of macros and traditional source code, from which compilable source code can be generated." Donald Knuth, 1984.

Wikipedia, 18/08/2020

[https://en.wikipedia.org/wiki/Literate\\_programming#Workflow](https://en.wikipedia.org/wiki/Literate_programming#Workflow)

# Introduction

What does it look like ?

The image shows two Jupyter Notebook windows side-by-side. The left window is a smaller preview or thumbnail of the main notebook. The right window is the full Jupyter interface.

**Title Bar:** jupyter Lorenz Differential Equations (autosaved) | Python 3

**Header:** File Edit View Insert Cell Kernel Help

**Toolbar:** Cell Toolbar: None

**Section Title:** Exploring the Lorenz System

In this Notebook we explore the [Lorenz system](#) of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

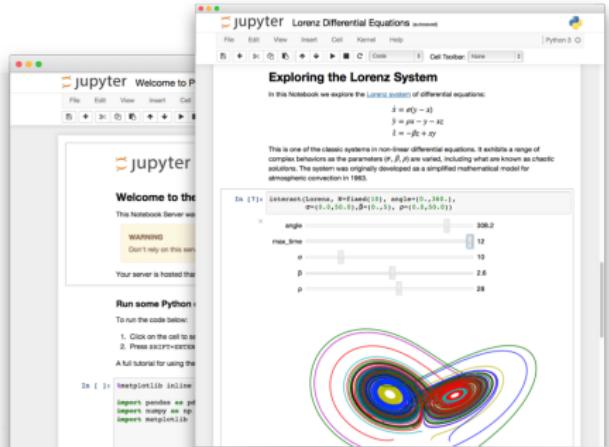
This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters ( $\sigma, \beta, \rho$ ) are varied, including what are known as chaotic solutions. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

**In [7]:** interact(Lorenz, N=fixed(10), angle=(0.,360.),  
sigma=(0.0,50.0),beta=(0.,5.),rho=(0.0,50.0))

angle: 308.2  
max\_time: 12  
 $\sigma$ : 10  
 $\beta$ : 2.6  
 $\rho$ : 28

**Figure:** A 3D plot of the Lorenz attractor, showing a double scroll pattern with three nested, spiraling trajectories in green, red, and blue.

# Introduction



Interactive programming interface allowing to combine both natural and computer languages.

In one file:

- Explanations
- Code
- Results
- Graphs and plots

# Introduction

Why using literate programming frameworks ?

Use cases:

- Day to day analyses
- Analysis reports
- Writing scientific articles

# Example of an article entirely written using a notebook

## File (on a repository)



## Published article



## Executable file



# Literate programming

This session :

- Markdown
- Rmarkdown / RStudio
- Jupyter

# Markup and markdown

## Definition

A markup language uses tags to define elements within a document.

Three different types and usage :

- Presentational (used by traditional word-processing systems)
  - ▶ Markup is invisible
- Procedural, provides instructions to process the text (e.g. TeX, PostScript)
  - ▶ Markup is visible and can be directly manipulated by the author.
- Descriptive, to label documents parts (e.g. LaTeX, HTML, XML...)
  - ▶ Emphasizes the document structure.

# Markdown language

Markdown is a Lightweight markup language.

Designed to be :

- easy to write using any generic text editor (plain-text-formatting syntax)
- easy to read in its raw form

# Markdown language

You've probably seen it already on GitHub (README), Wikipedia...

```
# Heading
```

```
## Sub-heading
```

```
### Another deeper heading
```

```
A [link](http://example.com).
```

```
Text attributes _italic_, *italic*, **bold**, `monospace`.
```

```
Bullet list:
```

- \* apples
- \* oranges
- \* pears

Github guide :

url <https://guides.github.com/features/mastering-markdown/>

# Literate programming

But how is this useful for literate programming?

When you want to weave both code (to be interpreted) and formatting information, you precisely need a lightweight language for the formatting part.

# The challengers

No need to hide, there are currently two main frameworks used in bioinformatics:  
RMarkdown and Jupyter

# RMarkdown

# RMarkdown

# RMarkdown

At the beginning, there was nothing.

# RMarkdown

At the beginning, there was nothing.

Then came Sweave.

Leisch, Friedrich (2002). "Sweave, Part I: Mixing R and LaTeX: A short introduction to the Sweave file format and corresponding R functions"

# RMarkdown

At the beginning, there was nothing.

Then came Sweave.

Leisch, Friedrich (2002). "Sweave, Part I: Mixing R and LaTeX: A short introduction to the Sweave file format and corresponding R functions"

And people saw that the path would be long...

# RMarkdown

knitr (2011)



"The knitr package was designed to be a transparent engine for dynamic report generation with R, solve some long-standing problems in Sweave, and combine features in other add-on packages into one package"

<https://yihui.org/knitr/>

# RMarkdown

## RMarkdown



"When you run render, R Markdown feeds the .Rmd file to knitr, which executes all of the code chunks and creates a new markdown (.md) document which includes the code and its output.

The markdown file generated by knitr is then processed by pandoc which is responsible for creating the finished format."

<https://rmarkdown.rstudio.com>

# RMarkdown

## RMarkdown

The screenshot shows the RStudio interface with the following details:

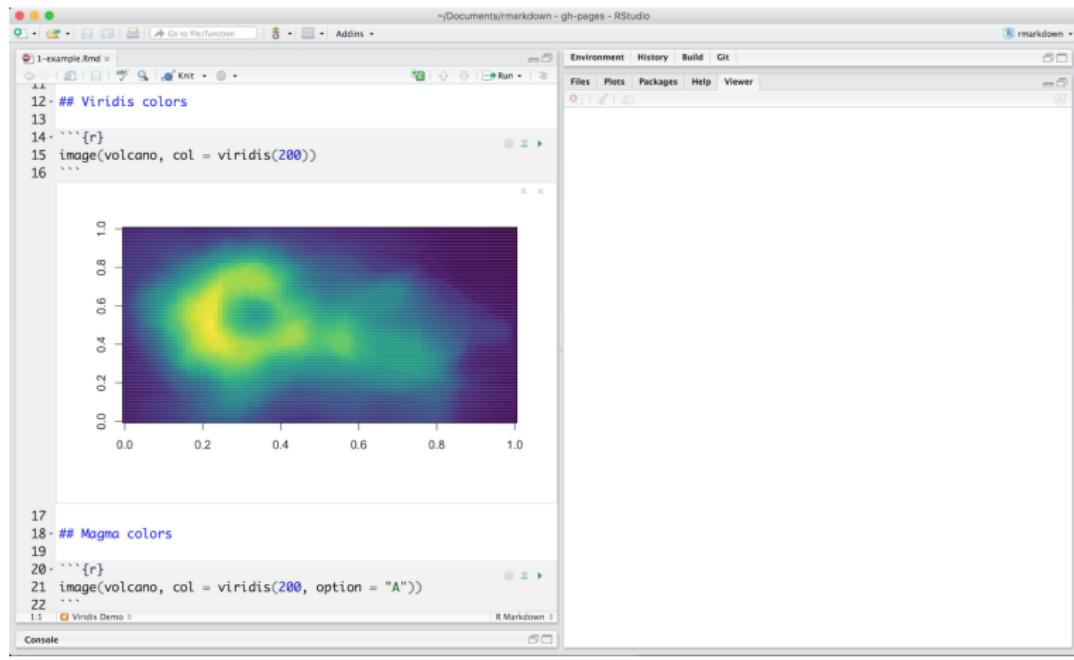
- Title Bar:** Shows the file name "1-example.Rmd" and the path "~Documents/rmarkdown - gh-pages - RStudio".
- Code Editor:** Displays the RMarkdown code. The code includes a YAML front matter block, several code blocks (```{r}```), and a narrative text block (10).

```
1---  
2 title: "Viridis Demo"  
3 output: html_document  
4---  
5  
6 ```{r include = FALSE}  
7 library(viridis)  
8```  
9  
10 The code below demonstrates two color palettes in the  
11 [viridis](https://github.com/sjmarnier/viridis) package. Each  
12 plot displays a contour map of the Maunga Whau volcano in  
13 Auckland, New Zealand.  
14- ## Viridis colors  
15- image(volcano, col = viridis(200))  
16-  
17- ## Magma colors  
18- image(volcano, col = viridis(200, option = "A"))  
19-  
20-  
21-  
22-  
23
```
- Environment Tab:** Shows tabs for Environment, History, Build, and Git.
- Files Tab:** Shows tabs for Files, Plots, Packages, Help, and Viewer.
- Console Tab:** Shows the status "R Markdown" and a "Console" tab.

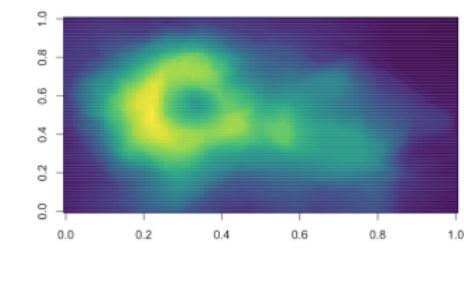
Integrated into RStudio, IDE for R.

# RMarkdown

## R Notebooks



The screenshot shows an RStudio interface with an R Notebook open. The notebook contains the following code:

```
## Viridis colors
```{r}
image(volcano, col = viridis(200))
```

## Magma colors
```{r}
image(volcano, col = viridis(200), option = "A")
```

```

The RStudio environment includes tabs for Environment, History, Build, and Git, and panels for Files, Plots, Packages, Help, and Viewer. The bottom of the screen shows the R Markdown interface and a console window.

## R Notebooks and more...

Markdown Basics

Output Formats

Notebooks

Slide Presentations

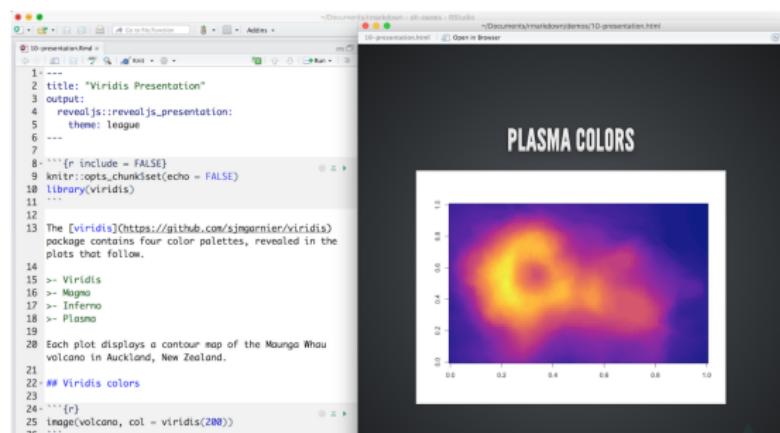
Dashboards

Websites

Interactive Documents

Cheat sheets

file below, which is available [here](#) on RStudio Cloud.



The screenshot shows the RStudio interface with two panes. The left pane displays an R Markdown code block:

```
1----
2 title: "Viridis Presentation"
3 output:
4   revealjs::revealjs_presentation:
5     theme: league
6 ---
7
8 ## {r include = FALSE}
9 knitr::opts_chunk$set(echo = FALSE)
10 library(viridis)
11 ...
12
13 The [viridis](https://github.com/sjmgarnier/viridis) package contains four color palettes, revealed in the plots that follow.
14
15 >- Viridis
16 >- Magma
17 >- Inferno
18 >- Plasma
19
20 Each plot displays a contour map of the Maunga Whau
volcano in Auckland, New Zealand.
21
22 ## Viridis colors
23
24 ## {r}
25 image(volcano, col = viridis(200))
26 ...
```

The right pane shows a presentation slide with the title "PLASMA COLORS" and a heatmap plot of a volcano contour map using the "Plasma" color palette from the viridis package.

# Jupyter

# Jupyter

# Jupyter

A bit of history...

- 2011 : IPython (interactive Python shell) with notebook functionalities
- 2014 : Spin-off project called Project Jupyter
- a non-profit, open-source project maintained by a strong Community
- "Jupyter will always be 100% open-source software, free for all to use and released under the liberal terms of the modified BSD license"
- A reference to the three core programming languages supported by Jupyter (Julia, Python and R)

<https://jupyter.org/>

# Jupyter

What can it do?

# Jupyter

What can it do?  
Everything (excepted coffee)

# Jupyter

But what is it exactly ?

# Jupyter

But what is it exactly ?

Web-based interactive computational environment.

# Jupyter

But what is it exactly ?

Web-based interactive computational environment.

- Web-based : client/server

# Jupyter

But what is it exactly ?

Web-based interactive computational environment.

- Web-based : client/server
- Interactive : notebook system

# Jupyter

But what is it exactly ?

Web-based interactive computational environment.

- Web-based : client/server
- Interactive : notebook system
- Computational environment : console, many kernels available...

# Jupyter

## Dashboard

The screenshot shows a Jupyter Notebook dashboard with the following interface elements:

- Toolbar:** Standard OS X-style toolbar with icons for file operations (New, Open, Save, Print, etc.).
- Title Bar:** Shows the URL "localhost" and a refresh icon.
- Header:** A navigation bar with tabs: "Files" (selected), "Running" (highlighted in blue), and "Clusters".
- File Tree:** A sidebar titled "File Tree" containing a list of files and directories:
  - data
  - dev
  - Exploratory Data Analytics.ipynb
  - Lights Out.ipynb
  - Welcome to Python.ipynb **Running Notebook** (highlighted in red)
- Actions:** Buttons for "Upload" and "New" (with dropdown).
- Status:** A "Running" status indicator next to the "Welcome to Python.ipynb" entry.
- Bottom Navigation:** A horizontal bar with icons for back, forward, search, and other navigation functions.
- Logos:** Logos for "ifb" (Institut de Biologie du Développement) and "FAIR\_Bioinfo" are visible in the bottom right corner.

# Jupyter

## Notebook editor

The screenshot shows a Jupyter Notebook interface running on a temporary server. The title bar reads "jupyter Welcome to Python (unsaved changes)". The toolbar includes standard file operations like New, Open, Save, and Print, along with CellToolbar and Toolbar buttons. The menu bar has options File, Edit, View, Insert, Cell, Kernel, Help, and Menubar. The status bar indicates "localhost" and "Python 3". A "Cell Mode Indicator" and "Kernel Indicator" are also present.

**jupyter** Welcome to Python (unsaved changes)

File Edit View Insert Cell Kernel Help **Menubar**

CellToolbar **Toolbar** Cell Mode Indicator | Kernel Indicator

**jupyter** Welcome to the Temporary Notebook (tmpnb) service!

This Notebook Server was launched just for you. It's a temporary way for you to try out a recent development version of the IPython/Jupyter notebook.

**WARNING**  
Don't rely on this server for anything you want to last - your server will be *deleted after 10 minutes of inactivity*.

Your server is hosted thanks to [Rackspace](#), on their on-demand bare metal servers, [OnMetal](#).

**Cell In Command Mode**

**Run some Python code!**

To run the code below:

1. Click on the cell to select it.
2. Press SHIFT+ENTER on your keyboard or press the play button (▶) in the toolbar above.

A full tutorial for using the notebook interface is available [here](#).

```
In [ ]: %matplotlib inline
import pandas as pd
import numpy as np
import matplotlib
```

 IFB Institut für Biochemie und Biophysik

# Jupyter

## Project Jupyter

- A non-profit, open-source project maintained by a strong Community
- Adopted by the biggest in the Cloud industry (Google, Microsoft, Amazon...)
- And financed by the biggest (Google, Microsoft, EU Horizon 2020 program, Alfred P. Sloan Foundation...)

Inside the Python community (snakemake, conda...)

Integration with GitHub since 2015 (renderer)

# Jupyter

Nbviewer : a static renderer for Jupyter notebooks



[JUPYTER](#) [FAQ](#)

# nbviewer

A simple way to share Jupyter Notebooks

Enter the location of a Jupyter Notebook to have it rendered here:

<https://nbviewer.jupyter.org/>



# Jupyter

Jupyter + Docker = binder



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a Zero-to-Binder tutorial in [Julia](#), [Python](#) or [R](#).

Build and launch a repository

GitHub repository name or URL

GitHub  GitHub repository name or URL

Git branch, tag, or commit

Path to a notebook file (optional)

Git branch, tag, or commit  Path to a notebook file (optional)  File

Copy the URL below and share your Binder with others:

Fill in the fields to see a URL for sharing your Binder.

<https://mybinder.org/>



# Jupyter

Since June 2019 : Jupyter Lab v1.0 (now v3.0.16)

The screenshot displays the Jupyter Lab interface with several open windows:

- File Explorer:** Shows a tree view of files and folders, including a "Notebooks" section.
- Code Editor:** An "In Depth: Linear Regression" notebook is open, containing Python code and text explaining linear regression.
- Terminal:** A terminal window shows command-line interactions for various languages (Python 3, C++11, C++14, C++17, Julia 1.1.0, juliabench (Python 3.7), R) and tools (Nanopipe).
- Output View:** Displays a scatter plot titled "Seattle Weather: 2012-2015" showing "Maximum Daily Temperature (F)" over time.
- Julia Myself:** A Julia notebook showing code for generating a 3D surface plot.
- python notebook:** A Python notebook showing code for solving the Lorenz system of differential equations.
- R:** An R notebook showing a scatter plot of Sepal.Length vs Sepal.Width.

# Conclusion ?

Who's the best?

# Conclusion ?

Who's the best?

It depends...

# Conclusion ?

Who's the best?

It depends...

- R analyses? Go for RMarkdown/RStudio
- R analyses for a publication ? Consider Jupyter with an R kernel

# Conclusion ?

Who's the best?

It depends...

- R analyses? Go for RMarkdown/RStudio
- R analyses for a publication ? Consider Jupyter with an R kernel
- Python analyses ? Why do you even ask...

# Practical session

## Savoir FAIRe

- Markdown
- Learn the structure of an Rmd file
- Turn a script into a notebook
- Extend the notebook with new functionalities
- This afternoon: Jupyter with the IFB cluster