

FAIR_bioinfo for bioinformaticians

Introduction to the tools of reproducibility in bioinformatics

C. Hernandez¹ T. Denecker¹ J. Sellier² G. Le Corguillé²
C. Toffano-Nioche¹

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

²IFB Core Cluster taskforce

Sept. 2020



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



Introduction

What does it look like ?

The image shows a Jupyter Notebook interface with the following content:

Exploring the Lorenz System

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

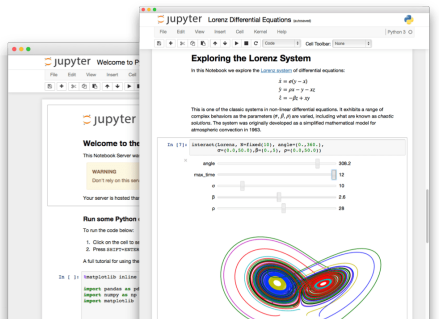
$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) 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
 σ : 10
 β : 2.6
 ρ : 28

Introduction



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

In one file:

- Explanations
- Code
- Results
- Graphs and plots

Introduction

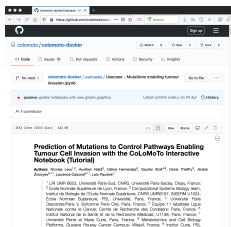
Why using literate programming frameworks ?

Use cases:

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

Example of an article entirely written using a notebook

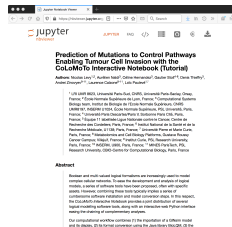
File (on a repository)



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 see 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 :

[urlhttps://guides.github.com/features/mastering-markdown/](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

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...



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



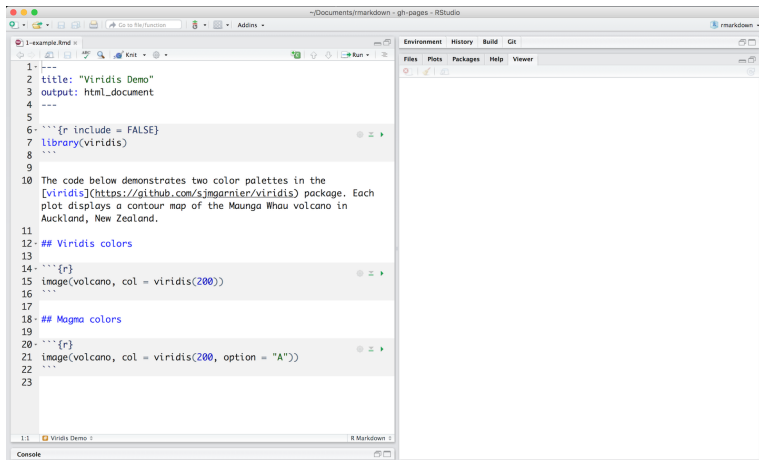
”When you run render, R Markdown feeds the .Rmd file to knitr, which executes all of the code chunks and creates a new markdown (.md) document which includes the code and its output.

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



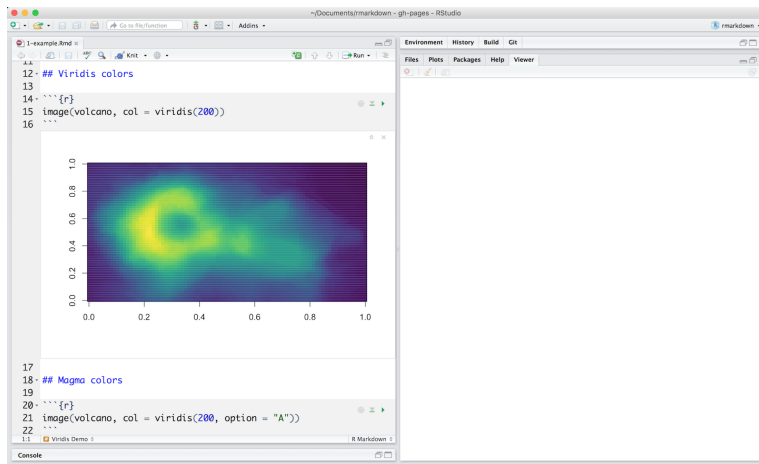
```
1 ---
2 title: "Viridis Demo"
3 output: html_document
4 ---
5
6 ```{r include = FALSE}
7 library(viridis)
8 ```
9
10 The code below demonstrates two color palettes in the
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each
12 plot displays a contour map of the Maunga Whau volcano in
13 Auckland, New Zealand.
14
15 ## Viridis colors
16 ```{r}
17 image(volcano, col = viridis(200))
18 ```
19
20 ## Magma colors
21 ```{r}
22 image(volcano, col = viridis(200, option = "A"))
23 ```
```

Integrated into RStudio, IDE for R.



RMarkdown

R Notebooks



R Notebooks and more...

Markdown Basics

Output Formats

Notebooks

Slide Presentations

Dashboards

Websites

Interactive Documents

Cheatsheets

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

The screenshot shows two windows from RStudio. The left window displays R code for a presentation slide:

```
1 <---
2 title: "Viridis Presentation"
3 output:
4 revealjs::revealjs_presentation:
5   theme: league
6 <---
7
8 ```{r include = FALSE}
9 knitr::opts_chunkset(echo = FALSE)
10 library(viridis)
11 ```
12
13 The [viridis](https://github.com/sjmgarnier/viridis)
14 package contains four color palettes, revealed in the
15 plots that follow.
16
17 >> Viridis
18 >> Magna
19 >> Inferno
20 >> Plasma
21
22 Each plot displays a contour map of the Maunga Whau
23 volcano in Auckland, New Zealand.
24
25 ## Viridis colors
26
27 ```{r}
28 image(volcano, col = viridis(200))
29 ```
```

The right window shows the rendered HTML slide presentation. The slide has a black background with the title "PLASMA COLORS" in white. Below the title is a contour plot of the Maunga Whau volcano, rendered using the Plasma color palette. The plot shows a central yellow/orange peak transitioning to red and then purple/blue at the edges. The x and y axes of the plot range from 0.0 to 1.0.

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

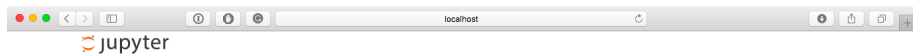


Jupyter

But what is it exactly ?

Web-based interactive computational environment.

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



Files **Running** Clusters

Select items to perform actions on them.

Upload New

<input type="checkbox"/>			File Tree	
<input type="checkbox"/>			data	
<input type="checkbox"/>			dev	
<input type="checkbox"/>			Exploratory Data Analytics.ipynb	
<input type="checkbox"/>			Lights Out.ipynb	
<input type="checkbox"/>			Welcome to Python.ipynb Running Notebook	Running


Notebook editor

localhost

jupyter Welcome to Python (unsaved changes) Python 3

File Edit View Insert Cell Kernel Help **Menubar** | Python 3

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](#).

Run some Python code! **Cell In Command Mode**

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
```

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)



Nbviewer : a static renderer for Jupyter notebooks



nbviewer

A simple way to share Jupyter Notebooks

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

`https://nbviewer.jupyter.org/`



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 ▾

Git branch, tag, or commit

Path to a notebook file (optional) File ▾

Copy the URL below and share your Binder with others:

<https://mybinder.org/>

The image displays a collage of Jupyter Lab interface elements:

- Top Left:** A file explorer showing a directory structure with folders like 'auto', 'images', 'Atan.jupyter', 'Cep.jupyter', 'Data.jupyter', 'Higgs.jupyter', and 'Julia.jupyter'. A 'Linear Regression.ipynb' file is selected.
- Top Center:** A code editor window titled 'In Depth: Linear Regression'. It contains introductory text about linear regression models and a code cell with kernel metadata for Python 3.
- Top Right:** A 'Launcher' window showing icons for Python 3, C++, C#, and C++T, along with a 'Console' window.
- Middle Right:** A plot titled 'Beetle Weather: 2012-2019' showing 'Maximum Daily Temperature (C)' on the y-axis and 'Date' on the x-axis. It includes a scatter plot and a bar chart for 'Number of Beetles'.
- Bottom Row:** Three smaller notebook windows:
 - Julia:** A plot titled 'Julia' showing 'wavelength (nm)' vs 'intensity'.
 - python notebook:** A code cell with mathematical equations: $\dot{x} = a(x - x_0)$, $\dot{y} = b(y - y_0) - c$, and $\dot{z} = -d + e \cdot y$. It also shows a code cell for solving a Lorenz system.
 - R:** A scatter plot titled 'R' showing a positive correlation between two variables.

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 ?

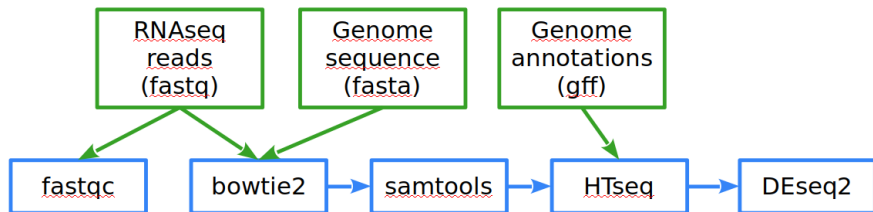
Who's the best?

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

Analysis workflow



green=input, blue=tool

fastqc control quality of the input reads

bowtie2 reads mapping on the genome sequence

samtools mapped reads selection & formatting

HTseq count table of mapped reads on genes (annotations)

DESeq2 statistical analysis: genes list having differential expression

Practical session

Savoir FAIRe

- Markdown
- Learn the structure of an Rmd file
- Turn a script into a notebook
- Extend the notebook with new functionalities
- Bonus: introduction to Jupyter