NTK
50°6'14.083"N, 14°23'26.365"E
Národní technická knihovna
National Library of Technology

Czech Academic and Research
Discovery Services
CARDS

Jupyter Book

Martin Schätz

24.3.2024

National Repository Platform for Research Data

Co-funded by
the European Union

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

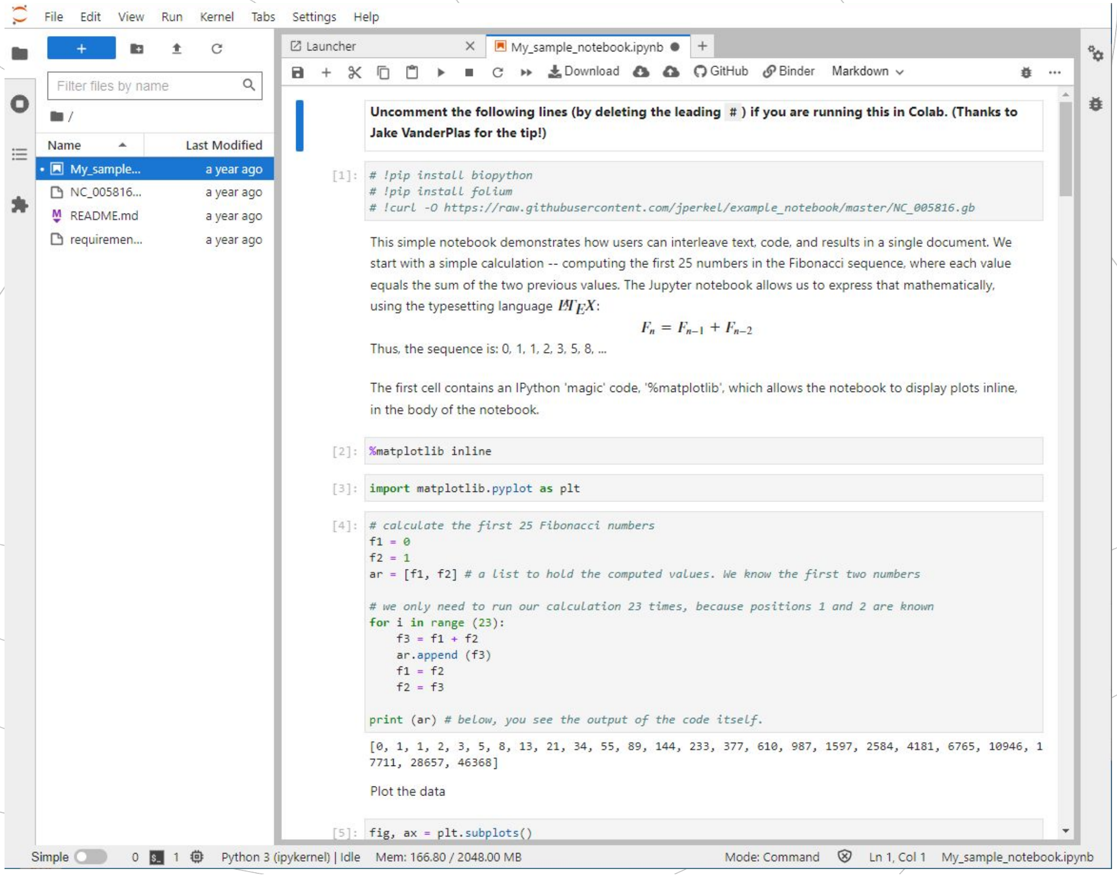
Jupyter Notebook?

Jupyter Notebook is an open-source web application to record, create and share computational documents. It is an **open document format** based on JSON. They contain a complete record of the user's sessions and include code, narrative text, equations, visualizations, and text.

It is go to Electronic Laboratory Notebook for many computer scientists.

- Writing procedures step-by-step using cells.
- Text using Latex syntax.
- Written in python, but extensible using kernels.

Jupyter Notebook?



Name	Last Modified
My_sample...	a year ago
NC_005816...	a year ago
README.md	a year ago
requiremen...	a year ago

Uncomment the following lines (by deleting the leading `#`) if you are running this in Colab. (Thanks to Jake VanderPlas for the tip!)

```
[1]: # !pip install biopython
     # !pip install folium
     # !curl -O https://raw.githubusercontent.com/jperkel/example_notebook/master/NC_005816.gb
```

This simple notebook demonstrates how users can interleave text, code, and results in a single document. We start with a simple calculation -- computing the first 25 numbers in the Fibonacci sequence, where each value equals the sum of the two previous values. The Jupyter notebook allows us to express that mathematically, using the typesetting language $\LaTeX$:

$$F_n = F_{n-1} + F_{n-2}$$

Thus, the sequence is: 0, 1, 1, 2, 3, 5, 8, ...

The first cell contains an IPython 'magic' code, '%matplotlib', which allows the notebook to display plots inline, in the body of the notebook.

```
[2]: %matplotlib inline
```

```
[3]: import matplotlib.pyplot as plt
```

```
[4]: # calculate the first 25 Fibonacci numbers
     f1 = 0
     f2 = 1
     ar = [f1, f2] # a list to hold the computed values. We know the first two numbers

     # we only need to run our calculation 23 times, because positions 1 and 2 are known
     for i in range (23):
         f3 = f1 + f2
         ar.append (f3)
         f1 = f2
         f2 = f3

     print (ar) # below, you see the output of the code itself.
```

```
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 1
7711, 28657, 46368]
```

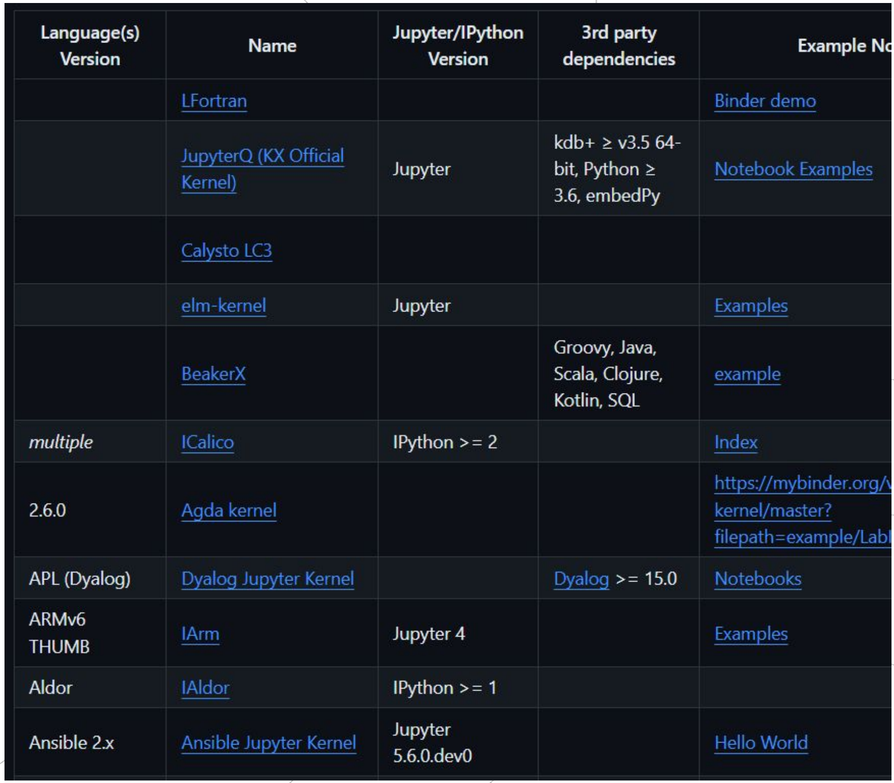
Plot the data

```
[5]: fig, ax = plt.subplots()
```

Kernels

A ‘kernel’ is a program that runs and introspects the user’s code. IPython includes a kernel for Python code, and people have written kernels for more than 40 other languages.

Kernel can be also python environment with specific python version and package version.



Language(s) Version	Name	Jupyter/IPython Version	3rd party dependencies	Example No
	LFortran			Binder demo
	JupyterQ (KX Official Kernel)	Jupyter	kdb+ ≥ v3.5 64-bit, Python ≥ 3.6, embedPy	Notebook Examples
	Calysto LC3			
	elm-kernel	Jupyter		Examples
	BeakerX		Groovy, Java, Scala, Clojure, Kotlin, SQL	example
multiple	ICalico	IPython >= 2		Index
2.6.0	Agda kernel			https://mybinder.org/v kernel/master? filepath=example/Labl
APL (Dyalog)	Dyalog Jupyter Kernel		Dyalog >= 15.0	Notebooks
ARMv6 THUMB	IArm	Jupyter 4		Examples
Aldor	IAldor	IPython >= 1		
Ansible 2.x	Ansible Jupyter Kernel	Jupyter 5.6.0.dev0		Hello World

Jupyter Book

Jupyter Book is an open-source tool for building publication-quality books and documents from computational material.

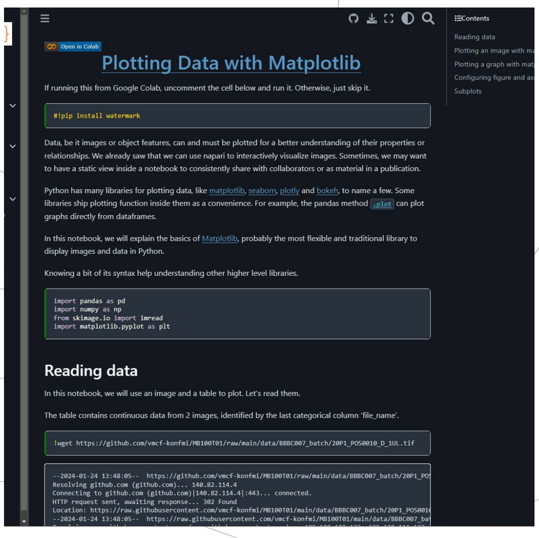
Jupyter Book allows users to

- write and share their content in **markdown** files or **Jupyter notebooks**,
- include computational elements (e.g., code cells) in either type,
- include rich syntax such as citations, cross-refs, and numbered equations, and
- using a simple command, run the embedded code cells, cache the outputs and convert this content into:
    - a web-based interactive book and
    - a publication-quality PDF.

With combination of version control, CI/CD or dockerization, this creates powerful reusable toolkit for Data Science.

Jupyter Book - Examples

Reusable Protocols


Contents
Reading data
Plotting an image with ma
Plotting a graph with mat
Configuring figure and ax
Subplots
Open in Colab

Plotting Data with Matplotlib

If running this from Google Colab, uncomment the cell below and run it. Otherwise, just skip it.

```
#!pip install watermark
```

Data, be it images or object features, can and must be plotted for a better understanding of their properties or relationships. We already saw that we can use napari to interactively visualize images. Sometimes, we may want to have a static view inside a notebook to consistently share with collaborators or as material in a publication.

Python has many libraries for plotting data, like matplotlib, seaborn, plotly and bokeh, to name a few. Some libraries ship plotting function inside them as a convenience. For example, the pandas method `.plot` can plot graphs directly from dataframes.

In this notebook, we will explain the basics of Matplotlib, probably the most flexible and traditional library to display images and data in Python.

Knowing a bit of its syntax help understanding other higher level libraries.

```
import pandas as pd
import numpy as np
from skimage.io import imread
import matplotlib.pyplot as plt
```

Reading data

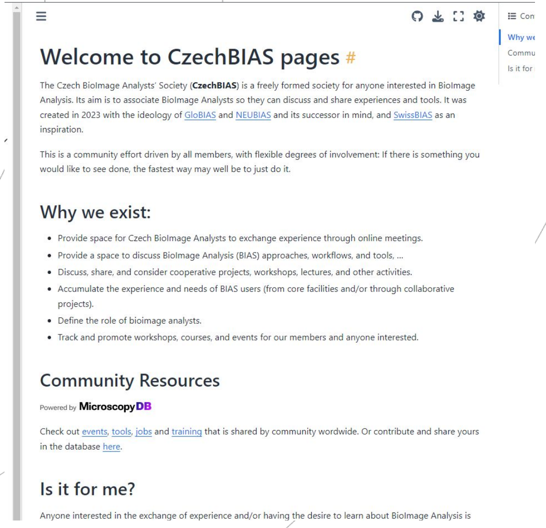
In this notebook, we will use an image and a table to plot. Let's read them.

The table contains continuous data from 2 images, identified by the last categorical column 'file_name'.

```
!wget https://github.com/vmcf-konfmi/MB100T01/raw/main/data/BBBC007_batch/20P1_POS0010_D_1UL.tif
```

```
--2024-01-24 13:48:05--  https://github.com/vmcf-konfmi/MB100T01/raw/main/data/BBBC007_batch/20P1_POS
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)|140.82.114.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/vmcf-konfmi/MB100T01/main/data/BBBC007_batch/20P1_POS001
--2024-01-24 13:48:05--  https://raw.githubusercontent.com/vmcf-konfmi/MB100T01/main/data/BBBC007_bat
```

Community Web



Welcome to CzechBIAS pages

The Czech BioImage Analysts' Society (**CzechBIAS**) is a freely formed society for anyone interested in BioImage Analysis. Its aim is to associate BioImage Analysts so they can discuss and share experiences and tools. It was created in 2023 with the ideology of GloBIAS and NEUBIAS and its successor in mind, and SwissBIAS as an inspiration.

This is a community effort driven by all members, with flexible degrees of involvement: If there is something you would like to see done, the fastest way may well be to just do it.

Why we exist:

- Provide space for Czech BioImage Analysts to exchange experience through online meetings.
- Provide a space to discuss BioImage Analysis (BIAS) approaches, workflows, and tools, ...
- Discuss, share, and consider cooperative projects, workshops, lectures, and other activities.
- Accumulate the experience and needs of BIAS users (from core facilities and/or through collaborative projects).
- Define the role of bioimage analysts.
- Track and promote workshops, courses, and events for our members and anyone interested.

Community Resources

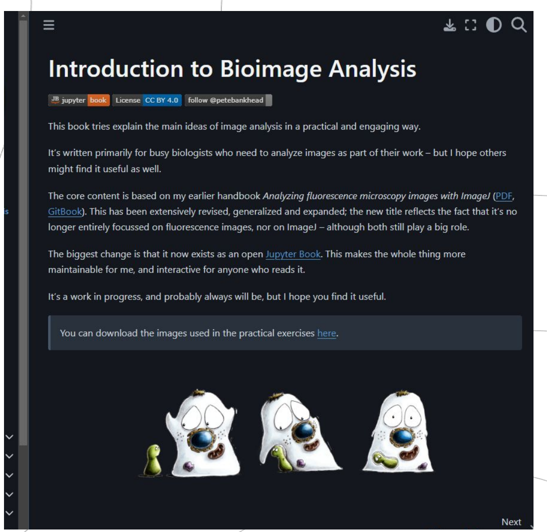
Powered by MicroscopyDB

Check out events, tools, jobs and training that is shared by community wordwide. Or contribute and share yours in the database here.

Is it for me?

Anyone interested in the exchange of experience and/or having the desire to learn about BioImage Analysis is

Books



Introduction to Bioimage Analysis

This book tries explain the main ideas of image analysis in a practical and engaging way.

It's written primarily for busy biologists who need to analyze images as part of their work – but I hope others might find it useful as well.

The core content is based on my earlier handbook *Analyzing fluorescence microscopy images with ImageJ* (PDF, GitBook). This has been extensively revised, generalized and expanded; the new title reflects the fact that it's no longer entirely focussed on fluorescence images, nor on ImageJ – although both still play a big role.

The biggest change is that it now exists as an open Jupyter Book. This makes the whole thing more maintainable for me, and interactive for anyone who reads it.

It's a work in progress, and probably always will be, but I hope you find it useful.

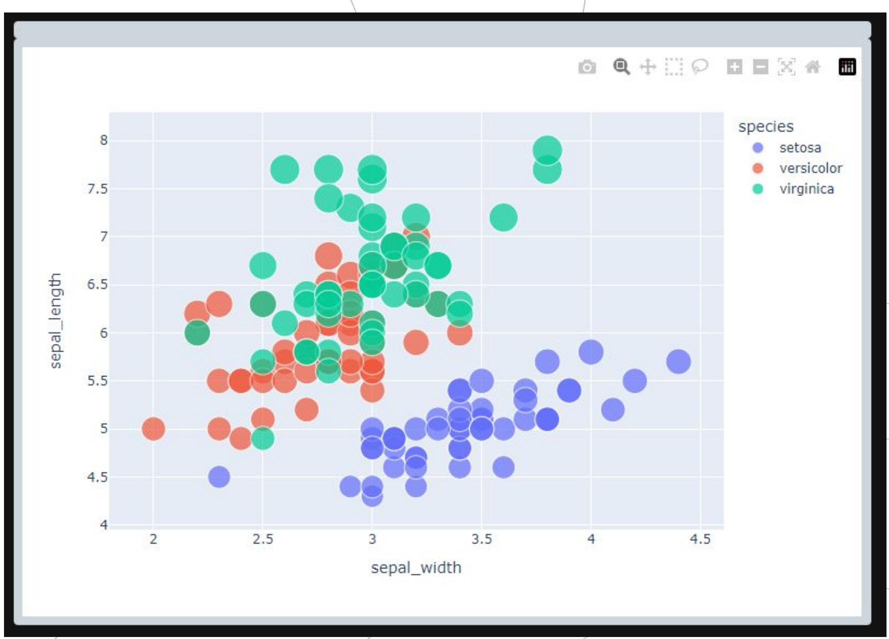
> You can download the images used in the practical exercises here.

Interactivity

Jupyter Notebook has support for many kinds of interactive outputs, including the ipywidgets ecosystem as well as many interactive visualization libraries. These are supported in Jupyter Book, with the right configuration.

Examples

Executable Books Gallery



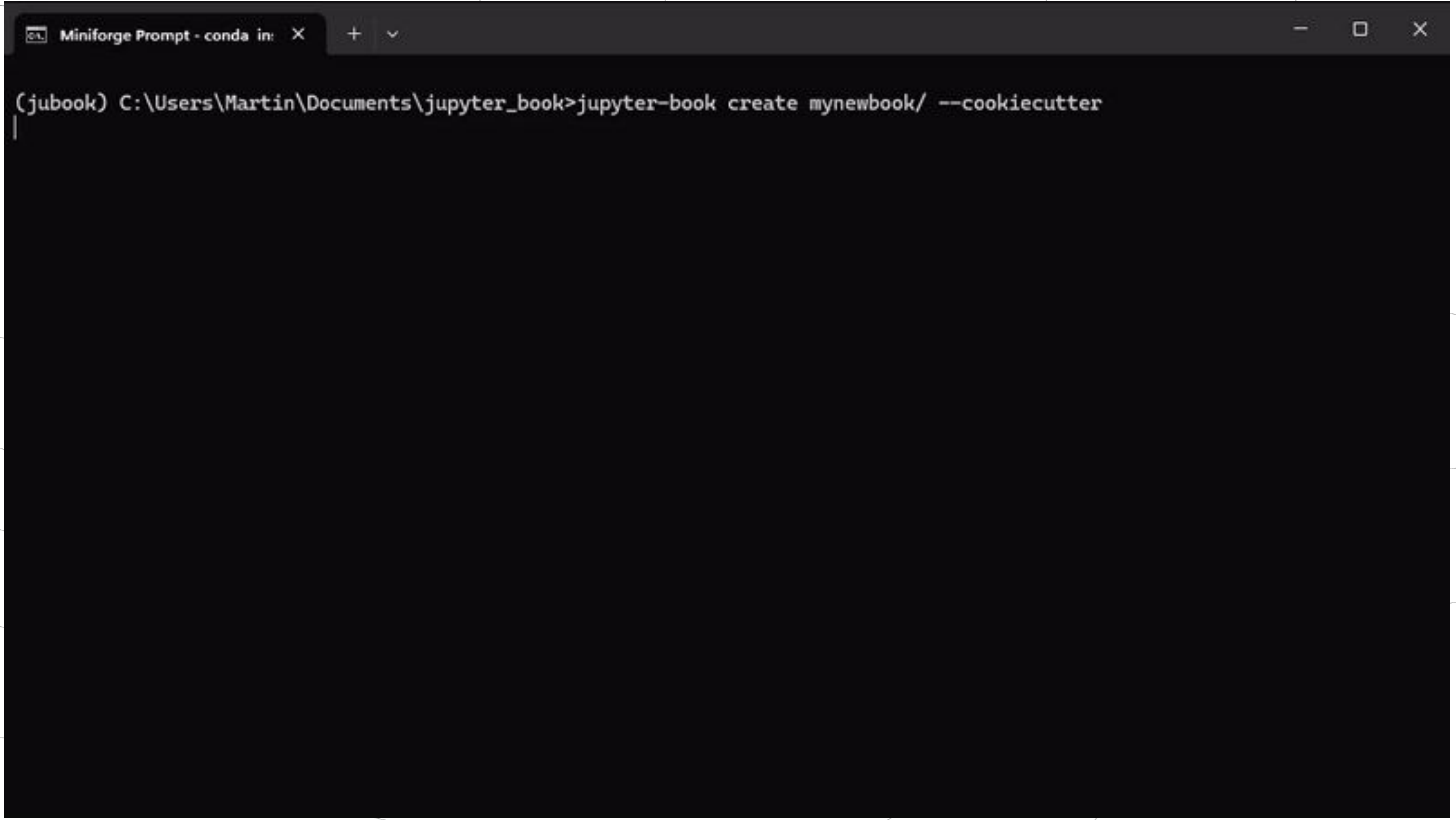
Cookiecutter - Developer's Toolkit

Create projects swiftly from cookiecutters (project templates) with this command-line utility. Ideal for generating Python package projects and more.

- **Cross-Platform**: Supports Windows, Mac, and Linux.
- **User-Friendly**: No Python knowledge required.
- **Versatile**: Compatible with Python 3.7 to 3.12.
- **Multi-Language Support**: Use templates in any language or markup format.

COOKIECUTTER

Cookiecutter



```
(jubook) C:\Users\Martin\Documents\jupyter_book>jupyter-book create mynewbook/ --cookiecutter
```

Versioning & Automatic Actions

Versioning

Whatever we do, backups and versioning is essential for development. While backing up and versioning of general data has various strategies. Text based data and code (or open formats) benefits from git, where we have option to track, comment and organize any change.

You can think about as MS Word track changes on steroids!

Automatic Actions

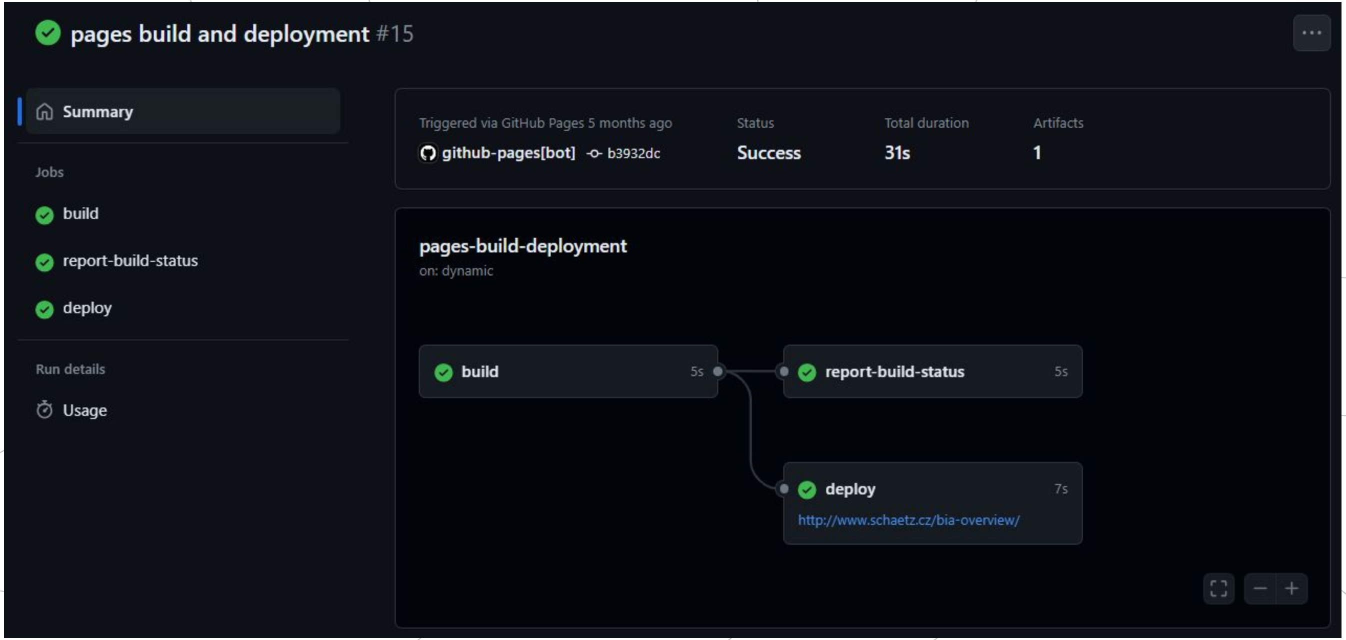
GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to **automate your build, test, and deployment pipeline**. You can create **workflows that build and test** every pull request to your repository, **or deploy** merged pull requests to production.

Versioning & Automatic Actions

Each proposed changed sent to GitHub is tested if it can be built.

Each pull request that is committed (changes in JB are accepted) gets automatically build and published to web!

This is called CI/CD



pages build and deployment #15

Triggered via GitHub Pages 5 months ago
github-pages[bot] b3932dc

Status: Success
Total duration: 31s
Artifacts: 1

pages-build-deployment
on: dynamic

http://www.schaetz.cz/bia-overview/

Interoperability - Binder & Colab

Binder is a free, open-source web service that packages Jupyter notebooks inside an executable container, which can be run within a web browser, no installation required. Colab allows users with Google accounts to execute Jupyter notebooks on the Google cloud. Code Ocean is a commercial code-execution and sharing service.

All of them are capable of launching instance from a link! Example

binder

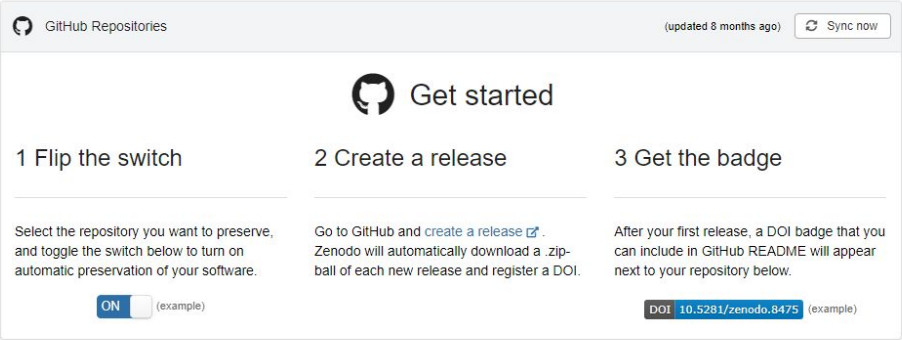
Binder is the go to solution when you want to work with:

- GitHub repositories,
- GitLab.com repositories,
- Published resources with DOI from Zenodo, Figsahre or Dataverse.

Publishing

Leveraging the “one-click” archiving of GitHub repository on Zenodo:

Whenever the repository contain code and data, text or collaborative Jupyter Book, we can automatically publish current content in Zenodo.



GitHub Repositories (updated 8 months ago) Sync now

Get started

1 Flip the switch

Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software.

ON (example)

2 Create a release

Go to GitHub and create a release. Zenodo will automatically download a .zip-ball of each new release and register a DOI.

3 Get the badge

After your first release, a DOI badge that you can include in GitHub README will appear next to your repository below.

DOI 10.5281/zenodo.8475 (example)

Let's explore the Quantitative Bioimaging paper website!

Senft, R. A., Diaz-Rohrer, B., Colarusso, P., Swift, L., Jamali, N., Jambor, H., Pengo, T., Brideau, C., Llopis, P. M., Uhlmann, V., Kirk, J., Gonzales, K. A., Bankhead, P., Evans 3rd, E. L., Giang, W., Haase, R., Costa Cruz, M., Schätz, M., Eliceiri, K. W., & Cimini, B. A. (2023). **Bioimagingguide.org - companion website to "A biologist's guide to planning and performing quantitative bioimaging experiments"** (2024.02.18.2). Zenodo. https://doi.org/10.5281/zenodo.10675761

Let’s write interactive Books!

References and Sources

- Community, E. (2020). Jupyter Book (Version v0.10). Zenodo. http://doi.org/10.5281/ZENODO.2561065
- https://jupyterbook.org/en/stable/interactive/interactive.html
- https://executablebooks.org/en/latest/gallery/
- https://wiki.metacentrum.cz/wiki/Jupyter_for_MetaCentrum_users#Binder
- https://github.com/jupyter/jupyter/wiki/Jupyter-kernels
- https://github.com/jperkel/example_notebook
- https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content
- https://cookiecutter.readthedocs.io/en/stable/
- https://github.com/jupyter/jupyter/wiki/Jupyter-kernels

Bonus: Practical Statistics in Medicine with R by Konstantinos I. Bougioukas, PhD