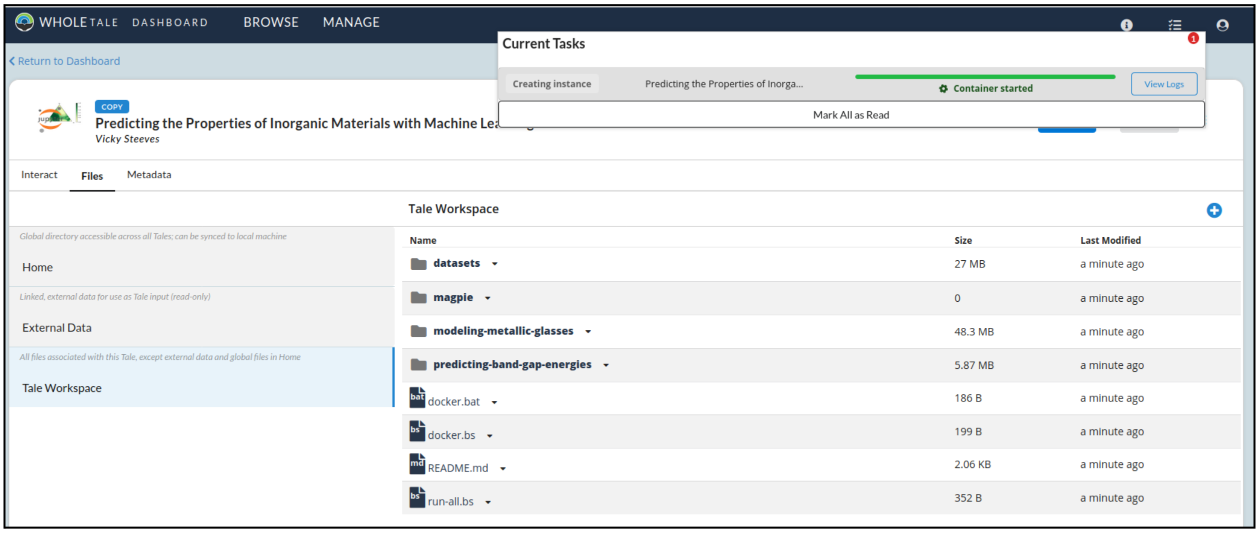
Appendix 3: Examples of Computational Reproducibility

This section will showcase some examples of how the software described in this chapter have been used to make research reproducible. Further examples can be found in the open book, The Practice of Reproducible Research, which comprises case studies and workflows for reproducibility across various disciplines.[1]

The first example of computationally reproducible research comes from a machine learning researcher, Logan Ward, using WholeTale to promote reuse of their materials. Their tale is meant to allow others to reproduce the materials in a "2016 paper on using machine learning to predict the properties of materials [...] the notebooks within this tale recreate the validation tests from the paper and how the models were used to discover new materials." Users who want to reuse this tale will have to either: a) create an account on WholeTale and copy it to their workspace to interact with it, or b) download the tale to their local computer and try to get it running with containers.



Current Tasks

Creating instance — Predicting the Properties of Inorga... — Container started

Predicting the Properties of Inorganic Materials with Machine Lea

Vicky Steeves

Global directory accessible across all Tales; can be synced to local machine

Home

Linked, external data for use as Tale input (read-only)

External Data

All files associated with this Tale, except external data and global files in Home

Tale Workspace

Tale Workspace

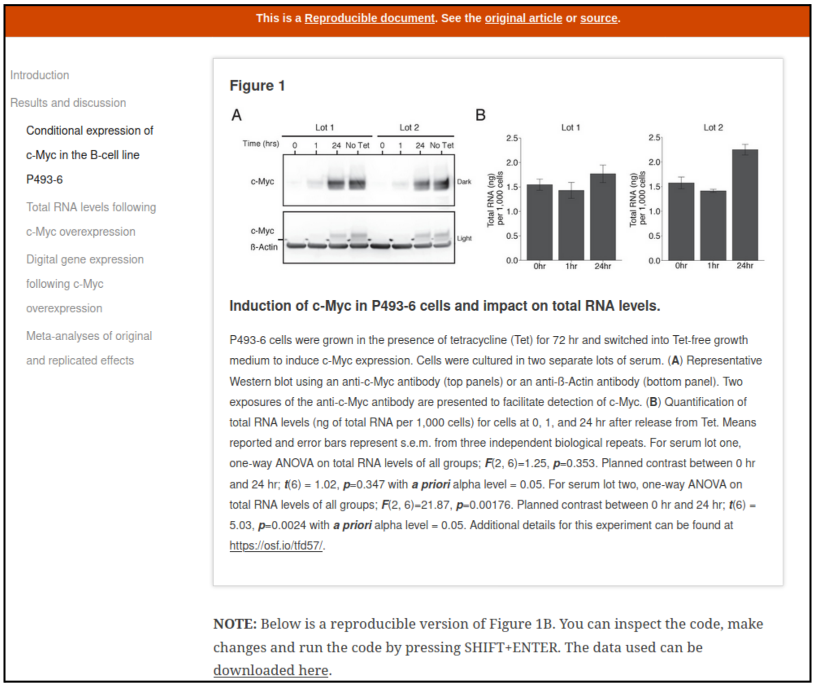
Name	Size	Last Modified
datasets	27 MB	a minute ago
magpie	0	a minute ago
modeling-metallic-glasses	48.3 MB	a minute ago
predicting-band-gap-energies	5.87 MB	a minute ago
docker.bat	186 B	a minute ago
docker.bs	199 B	a minute ago
README.md	2.06 KB	a minute ago
run-all.bs	352 B	a minute ago

Rerunning the Ward tale in my WholeTale account successfully.

The next example is from the biological sciences, where Lewis et al. used the e-Life Journals reproducible document stack (RDS) to provide an interactive version of their

[1] Justin Kitzes, Daniel Turek, and Fatima Deniz, eds., *The Basic Reproducible Workflow Template*. The Practice of Reproducible Research (Oakland, CA: University of California Press, 2018), https://www.practicereproducibleresearch.org/core-chapters/3-basic.html#.

paper, to allow others to directly rerun the code with original data that was used for analysis and visualizations.[2] The e-Life RDS is based on Stencila, a tool meant to introduce reproducibility features (such as updating dependencies in real-time between code cells as you change data or code) to everyday research tools (like Jupyter notebooks), and Docker, to keep the original computing environment. The code and data are then linked and able to be downloaded and explored by readers in real-time, augmenting their reading experience and allowing for open post-publication peer review.



This is a Reproducible document. See the original article or source.

Introduction
Results and discussion
Conditional expression of c-Myc in the B-cell line P493-6
Total RNA levels following c-Myc overexpression
Digital gene expression following c-Myc overexpression
Meta-analyses of original and replicated effects
Figure 1
A
Lot 1
Lot 2
Time (hrs)
0
1
24
No Tet
c-Myc
Dark
c-Myc
ß-Actin
Light
B
Lot 1
Lot 2
Total RNA (ng) per 1,000 cells
0.0
0.5
1.0
1.5
2.0
2.5
0hr
1hr
24hr

Induction of c-Myc in P493-6 cells and impact on total RNA levels.

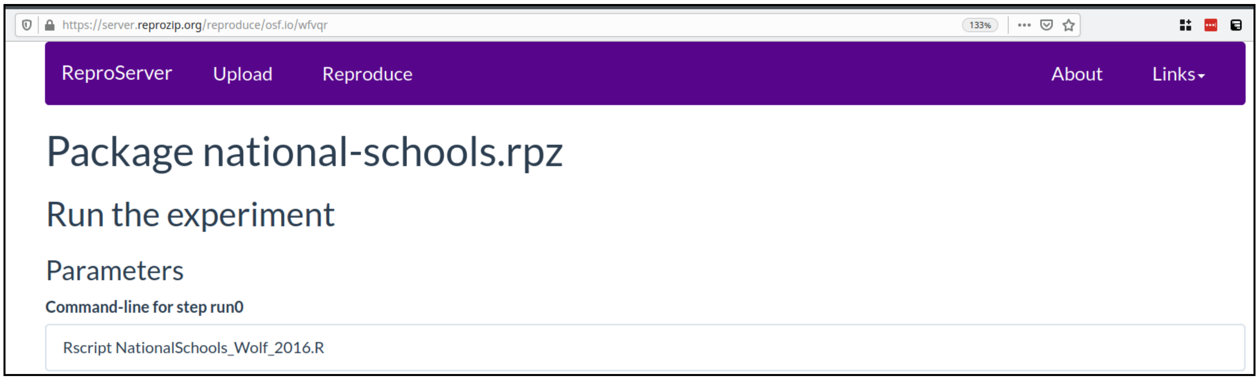
P493-6 cells were grown in the presence of tetracycline (Tet) for 72 hr and switched into Tet-free growth medium to induce c-Myc expression. Cells were cultured in two separate lots of serum. (**A**) Representative Western blot using an anti-c-Myc antibody (top panels) or an anti-ß-Actin antibody (bottom panel). Two exposures of the anti-c-Myc antibody are presented to facilitate detection of c-Myc. (**B**) Quantification of total RNA levels (ng of total RNA per 1,000 cells) for cells at 0, 1, and 24 hr after release from Tet. Means reported and error bars represent s.e.m. from three independent biological repeats. For serum lot one, one-way ANOVA on total RNA levels of all groups; $F(2, 6)=1.25$, $p=0.353$. Planned contrast between 0 hr and 24 hr; $t(6) = 1.02$, $p=0.347$ with ***a priori*** alpha level = 0.05. For serum lot two, one-way ANOVA on total RNA levels of all groups; $F(2, 6)=21.87$, $p=0.00176$. Planned contrast between 0 hr and 24 hr; $t(6) = 5.03$, $p=0.0024$ with ***a priori*** alpha level = 0.05. Additional details for this experiment can be found at https://osf.io/tfd57/.

NOTE: Below is a reproducible version of Figure 1B. You can inspect the code, make changes and run the code by pressing SHIFT+ENTER. The data used can be downloaded here.

A screenshot of the Lewis et. al. interactive paper.

Another example comes from digital humanities, where Nick Wolf made the materials available for his 2015 Heaney Lecture, “National School System and the Irish Language”.[3] Look for the published essay: "The National School System and the Irish Language in the Nineteenth Century".[4] Wolf used ReproZip to make a reproducible research compendium of the R scripts that he wrote to analyze and visualize historical education data from Ireland.

[2] L Michelle Lewis et al., “Replication Study: Transcriptional Amplification in Tumor Cells with Elevated c-Myc,” ed. Michael R Green, *ELife* 7 (January 9, 2018): e30274, https://doi.org/10.7554/eLife.30274.
[3] Nicholas Wolf, “National School System and the Irish Language Heaney Lecture 2015,” April 12, 2016, https://doi.org/10.17605/OSF.IO/PGK8V.
[4] Nicholas Wolf, “The National-School System and the Irish Language in the Nineteenth Century,” in Schools and Schooling, 1650–2000, 2017, 208, https://www.fourcourtspress.ie/books/2017/schools-and-schooling/.



Package national-schools.rpz

Run the experiment

Parameters

Command-line for step run0

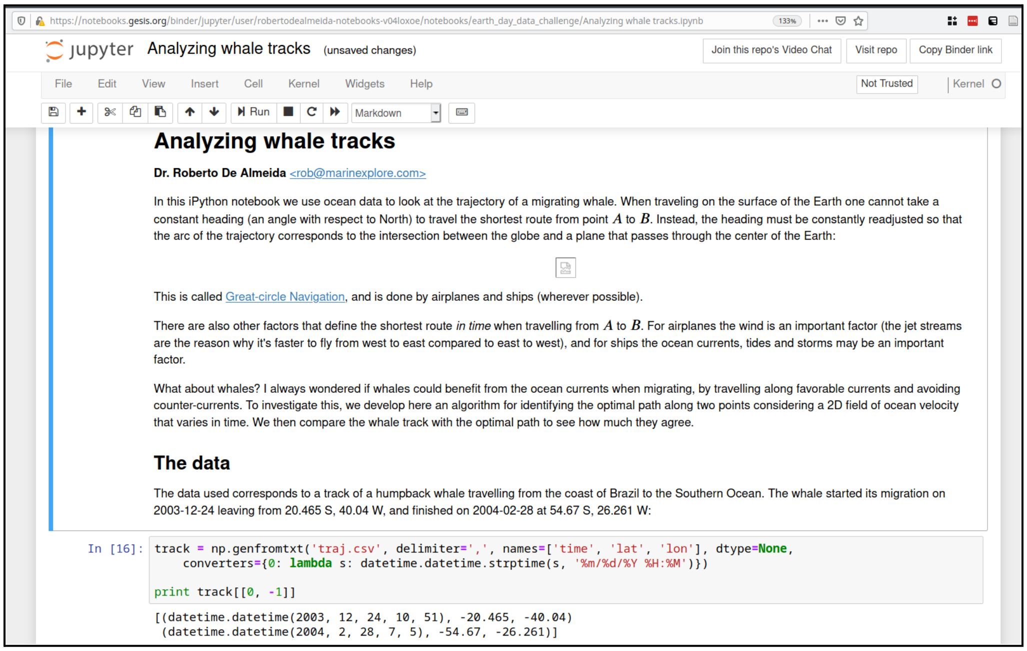
Rscript NationalSchools_Wolf_2016.R

A screenshot of unpacking Wolf’s RPZ bundle with ReproServer: https://server.reprozip.org/reproduce/osf.io/wfvqr. The results can be found at: https://server.reprozip.org/results/fyjog.

Wolf was able to package his research with two commands: `reprozip trace R Rscript NationalSchools_Wolf_2016.R` and `reprozip pack national-schools.rpz`. The RPZ bundle was then uploaded to the Open Science Framework (osf.io/wfvqr), where it can be either downloaded by secondary users for local interaction, or unpacked with ReproServer in browser for quick reproduction and inspection (see screenshot above).

The next example comes from earth science and mammalogy, using Jupyter notebooks with Binder for reproducibility. This notebook, *Analyzing whale tracks* by Dr. Roberto De Almeida looks at ocean data to track the trajectories of migrating whales. He wanted to see if whales could benefit from the ocean currents when migrating across the world.[5]

[5] Rob De Almeida, “Analyzing Whale Tracks,” EarthPy, September 23, 2013, http://earthpy.org/analyzing-whale-tracks.html.


https://notebooks.gesis.org/binder/jupyter/user/robertodealmeida-notebooks-v04loxoe/notebooks/earth_day_data_challenge/Analyzing whale tracks.ipynb
133%
jupyter
Analyzing whale tracks
(unsaved changes)
Join this repo's Video Chat
Visit repo
Copy Binder link
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Kernel
Run
Markdown

Analyzing whale tracks

Dr. Roberto De Almeida <rob@marinexplore.com>

In this iPython notebook we use ocean data to look at the trajectory of a migrating whale. When traveling on the surface of the Earth one cannot take a constant heading (an angle with respect to North) to travel the shortest route from point A to B. Instead, the heading must be constantly readjusted so that the arc of the trajectory corresponds to the intersection between the globe and a plane that passes through the center of the Earth:

This is called Great-circle Navigation, and is done by airplanes and ships (wherever possible).

There are also other factors that define the shortest route *in time* when travelling from A to B. For airplanes the wind is an important factor (the jet streams are the reason why it's faster to fly from west to east compared to east to west), and for ships the ocean currents, tides and storms may be an important factor.

What about whales? I always wondered if whales could benefit from the ocean currents when migrating, by travelling along favorable currents and avoiding counter-currents. To investigate this, we develop here an algorithm for identifying the optimal path along two points considering a 2D field of ocean velocity that varies in time. We then compare the whale track with the optimal path to see how much they agree.

The data

The data used corresponds to a track of a humpback whale travelling from the coast of Brazil to the Southern Ocean. The whale started its migration on 2003-12-24 leaving from 20.465 S, 40.04 W, and finished on 2004-02-28 at 54.67 S, 26.261 W:

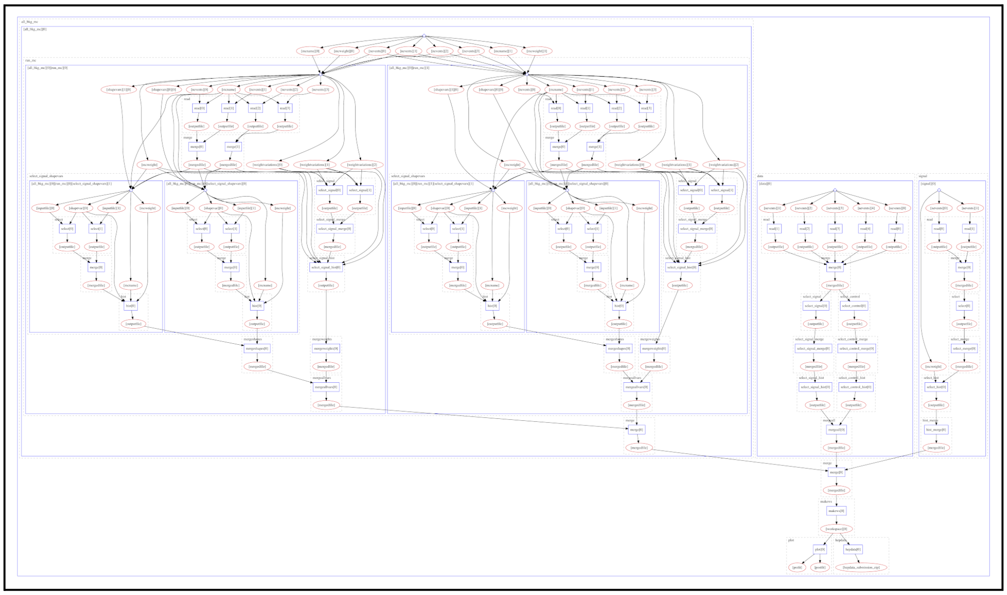
```
In [16]: track = np.genfromtxt('traj.csv', delimiter=',', names=['time', 'lat', 'lon'], dtype=None,
             converters={0: lambda s: datetime.datetime.strptime(s, '%m/%d/%Y %H:%M')})

         print track[[0, -1]]

         [(datetime.datetime(2003, 12, 24, 10, 51), -20.465, -40.04)
          (datetime.datetime(2004, 2, 28, 7, 5), -54.67, -26.261)]
```

A screenshot of the analyzing whale tracks Jupyter notebook running in Binder: https://gesis.mybinder.org/binder/v2/gh/robertodealmeida/notebooks/21221931f71eeec1bd35ed2dbecfc08869ffbeb3.

People can interact with the GitHub repository of Jupyter notebooks locally, by installing Jupyter notebooks and all the requisite dependencies (e.g. the correct Python version, and the correct Python library versions). They can also interact with the notebooks in Binder, to allow for simpler reproducibility in-browser. Users can interact with the notebooks with the same flexibility as if it was their local computers, re-executing and editing code, adding their own data, importing and exporting files, etc. These sandboxes do not persist, but instead offer a great way to instantly replay research during the reading or reviewing process.

The final example comes from high energy physics, where the REANA team created an example reproducible analysis pipeline of ATLAS data.[6] The workflow that they made reproducible with REANA emulates a "Beyond Standard Model (BSM) search as performed in collider particle physics." This involves reading in observed data, fitting it against a statistical model, and computing the upper limit on the signal strength of the BSM project (the main output).

[6] Diego Rodriguez et al., Reanahub/Reana-Demo-Bsm-Search, Python (2018; repr., REANA, 2020), https://github.com/reanahub/reana-demo-bsm-search.



The workflow that was made reproducible with REANA. I would not want to manually recreate that!

To create their reproducible analysis pipeline, they have to create:

> "[...] 'runnable recipes' addressing (1) where is the input data, (2) what software was used to analyse the data, (3) which computing environments were used to run the software and (4) which computational workflow steps were taken to run the analysis. This will permit instantiation of the analysis on the computational cloud and run the analysis to obtain (5) output results."[7]

The authors then put together into a reana.yml file which configures the analysis structure with the correct computational pipeline steps, inputs, parameters, dependencies, and code. It can then be deployed to a REANA server, one of which is hosted by CERN for use. This one is hard to reproduce without domain knowledge, or at least serious computational know-how.

Different reproducibility tools offer different functionality, which appeals to disciplines with varying norms. These examples offer some examples of how a few disciplines have used reproducibility tools to allow others to verify, extend, and interact with their work. By walking the walk, the authors above have provided great examples to follow in terms of making research reproducibly accessible to all.

[7] Rodriguez et al.

References

De Almeida, Rob. “Analyzing Whale Tracks.” *EarthPy*, September 23, 2013. http://earthpy.org/analyzing-whale-tracks.html.

Kitzes, Justin, Daniel Turek, and Fatima Deniz, eds. *The Basic Reproducible Workflow Template* The Practice of Reproducible Research. Oakland, CA: University of California Press, 2018. https://www.practicereproducibleresearch.org/core-chapters/3-basic.html#.

Lewis, L Michelle, Meredith C Edwards, Zachary R Meyers, C Conover Talbot Jr, Haiping Hao, David Blum, and Reproducibility Project: Cancer Biology. “Replication Study: Transcriptional Amplification in Tumor Cells with Elevated c-Myc.” Edited by Michael R Green. *ELife* 7 (January 9, 2018): e30274. https://doi.org/10.7554/eLife.30274.

Rodriguez, Diego, Lukas Heinrich, Maciulaitis Rokas, Tibor Simko, and Jan Okraska. Reanahub/Reana-Demo-Bsm-Search. Python. 2018. Reprint, REANA, 2020. https://github.com/reanahub/reana-demo-bsm-search.

Wolf, Nicholas. “National School System and the Irish Language Heaney Lecture 2015,” April 12, 2016. https://doi.org/10.17605/OSF.IO/PGK8V.

———. “The National-School System and the Irish Language in the Nineteenth Century.” *In Schools and Schooling*, 1650–2000, 208, 2017. https://www.fourcourtspress.ie/books/2017/schools-and-schooling/.