Beyond Prediction

Explanatory and Transparent Data Science

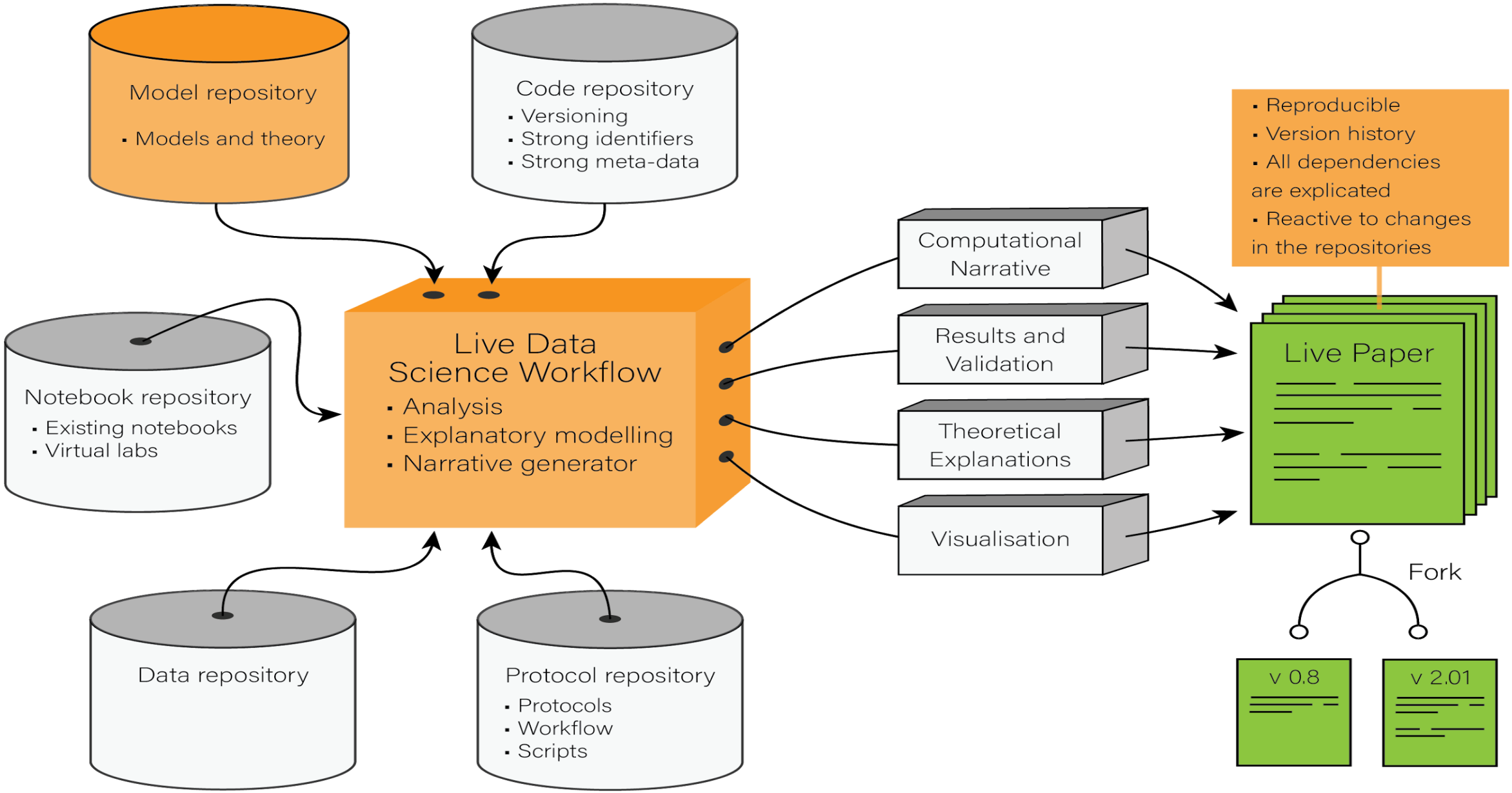
‘Live’ Data Science

When all the steps in the process of analysis -- from data discovery to application -- are made transparent, auditable and reactive to change (via continuous automated integration of new data, models and methods) we close the gap between doing research and its effective and timely communication.

Traditional media for communicating science (e.g., research journals) are fossilized objects that often contain errors, ambiguity and are disconnected from the original scientific process.

We are developing workflows that support reactive, dependency-based computations to facilitate truth maintenance in the face of changes to both data and methods.

Live Data Science Workflow



Meta Data Science: Some Areas of Contributions

Input and sources

Data is collected from various sources

Data processing

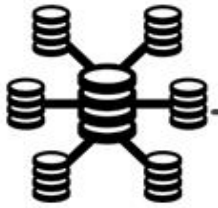
Data is ingested, cleansed, and stored

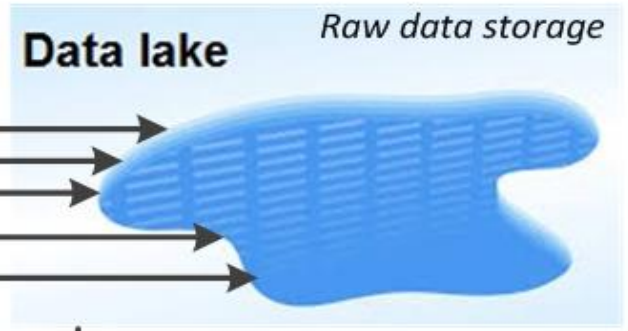
Application-driven schema transformation

Data quality requirements of applications drive the schema normalization process

Different designs fit for different purposes

Data in application schemata are fit for purpose by design





(1)

Cleansing

(2)

(3)

(3)

Stencila

https://stenci.la

Stencila

Executable document pipelines

Author, collaborate, and publish beautiful interactive documents on an open source web platform.

Sign up for free →

LAUNCH DEMO

Run Document

Source

Welcome to a new ERA of reproducible publishing

Emmy Tsang, Giuliano Maciocci

New open-source technology lets eLife authors publish Executable Research Articles that treat live code and data as first-class citizens.

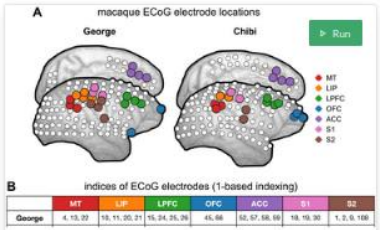
Since 2017, we have been working on the concept of computationally reproducible papers. The open-source suite of tools that started life as the Reproducible Document Stack is now live on eLife as ERA, the **Executable Research Article**, delivering a truly web-native format for taking published research to a new level of transparency, reproducibility and interactivity.

Projects : Stencila

https://hub.stenci.la/projects/

70%

Stencila Projects Organizations Pricing Sign in Sign up

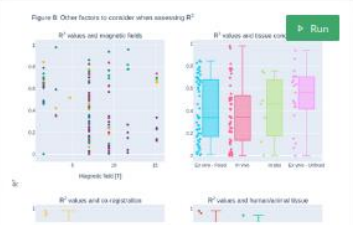


Run

Neuronal timescales are functionally dynamic and shaped by cortical microarchitecture

Complex cognitive functions such as working memory and decision-making require information maintenance over seconds to years, from transient sensory stimuli to long-term contextual cues. While theoretical accounts predict the emergence of a corresponding hierarchy of neuronal timescales, direct electrophysiological evidence acr…

elife



Run

JROST Lightning Talk Demo

A demonstration project based on Matteo Mancini et al's paper. It features a Jupyter Notebook pulled from https://github.com/matteomancini/myelin-meta-analysis with extensive interactive Plotly visualizations created using both R and Python. This project is mainly a test for a forthcoming Executable Research Article to be publi…

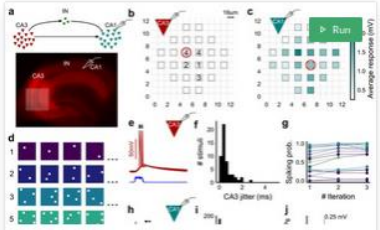
stencila

Run

epitopepredict: A tool for integrated MHC binding prediction

A key step in the cellular adaptive immune response is the presentation of antigen to T cells. During this process short peptides processed from self or foreign proteins may be presented on the surface bound to MHC molecules for binding to T cell receptors. Those that bind and activate an immune response are called epitopes. Co…

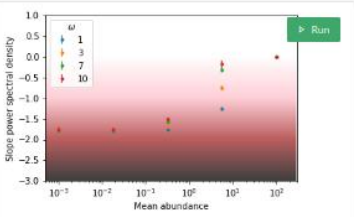
gigabyte



Run

Precise excitation-inhibition balance controls gain and timing in the hippocampus

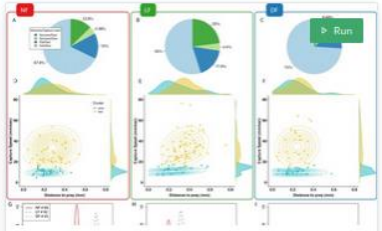
Excitation-inhibition (EI) balance controls excitability,



Run

Stochastic logistic models reproduce experimental time series of microbial communities

We analyze properties of experimental microbial time



Run

Learning steers the ontogeny of an efficient hunting sequence in zebrafish larvae

Goal-directed behaviors may be poorly coordinated in

Project snapshot : Stencil

https://hub.stenci.la/elife/article-61277/snapshots/71

Stencila Projects Organizations Pricing Sign in Sign up

Neuronal timescales are functionally dynamic and shaped by cortical microarchitecture

Files
Sources
Snapshots

Snapshot #71

Download Open

nokome · 2 months, 3 weeks ago · Finished

Preview

A preview of how your snapshot will appear to readers.

RUN DOCUMENT SOURCE

Neuronal timescales are functionally dynamic and shaped by cortical microarchitecture

Richard Gao, Ruud L van, den, Brink, Thomas Pfeffer, Bradley Voytek

Department of Cognitive Science, University of California, San Diego, La Jolla, United States; Section Computational Cognitive Neuroscience, Department of Neurophysiology and Pathophysiology, University Medical Center Hamburg-Eppendorf, Hamburg, Germany; Center for Brain and Cognition, Computational Neuroscience Group, Universitat Pompeu Fabra, Barcelona, Spain; Halıcıoğlu Data Science Institute, University of California, San Diego, La Jolla, United States; Neurosciences Graduate Program, University of California, San Diego, La Jolla, United States; Kavli Institute for Brain and Mind, University of California, San Diego, La Jolla, United States

Nov 23, 2020

Abstract

Complex cognitive functions such as working memory and decision-making require information maintenance over seconds to years, from transient sensory stimuli to long-term contextual cues. While theoretical accounts predict the emergence of a corresponding hierarchy of neuronal timescales, direct electrophysiological evidence across the human cortex is lacking. Here, we infer neuronal timescales from invasive intracranial recordings. Timescales increase along the principal sensorimotor-to-association axis across the entire human cortex, and scale with single-unit timescales within macaques. Cortex-wide transcriptomic analysis shows direct alignment between timescales and expression of excitation- and inhibition-related genes, as well as genes specific to voltage-gated transmembrane ion transporters. Finally, neuronal timescales are functionally dynamic: prefrontal cortex timescales expand during working memory maintenance and predict individual performance, while cortex-wide timescales compress with aging. Thus, neuronal timescales follow

Project snapshot : Stencil

https://hub.stenci.la/gigabyte/epitopepredict/snapshots/18

Stencila Projects Organizations Pricing Sign in Sign up

epitopepredict: A tool for integrated MHC binding prediction

Files
Sources
Snapshots

Snapshot #18

Download Open

nokome · 9 months ago · Finished

Preview

A preview of how your snapshot will appear to readers.

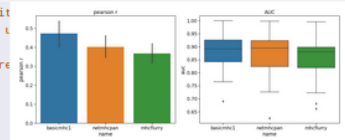
Run Document Source

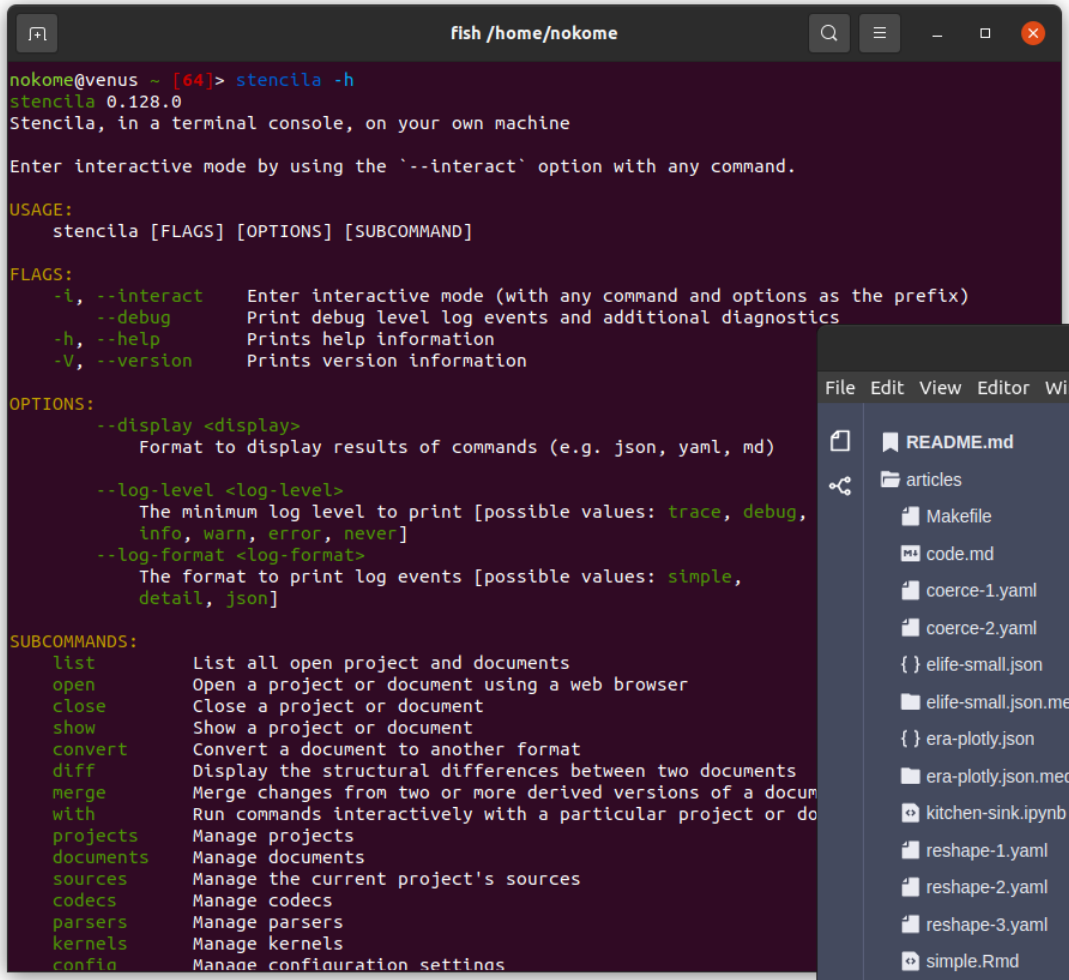
In practical use, this predictor can be run directly from the API or command line without installing any other program. Models are trained once as needed for each allele/length combination using the current installed versions of scikit-learn and joblib. Once trained, each model is saved and can be re-used. Training only takes a matter of seconds for each model.

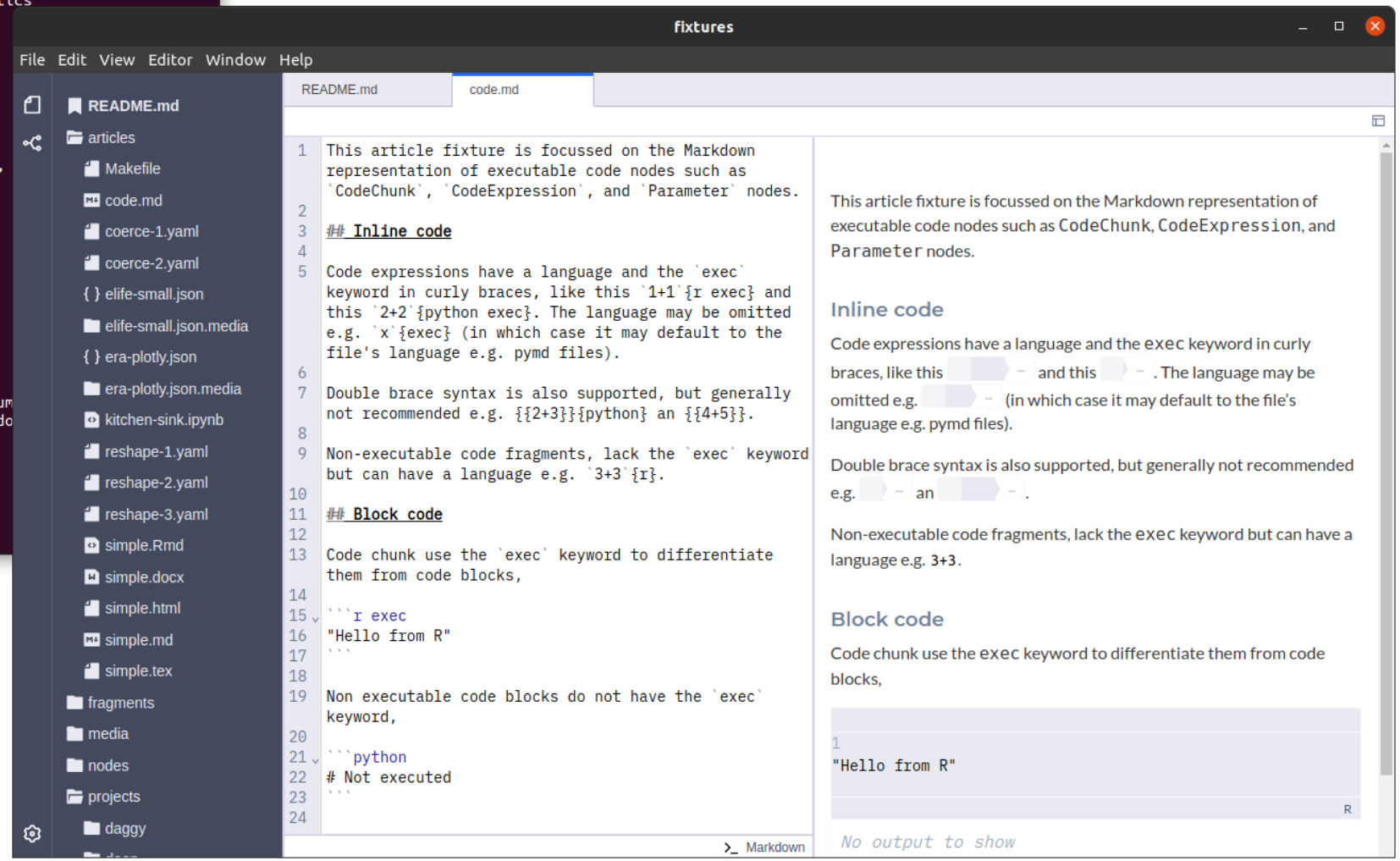
Figure 1

Performance of the basicmhc1 predictor compared to netMHCpan and MHCflurry for 40 human alleles. (a) Mean Pearson r and (b) mean AUC scores over all alleles. Only alleles with evaluation data for over more than 200 peptides were used. This test dataset used 9-mer peptides only.

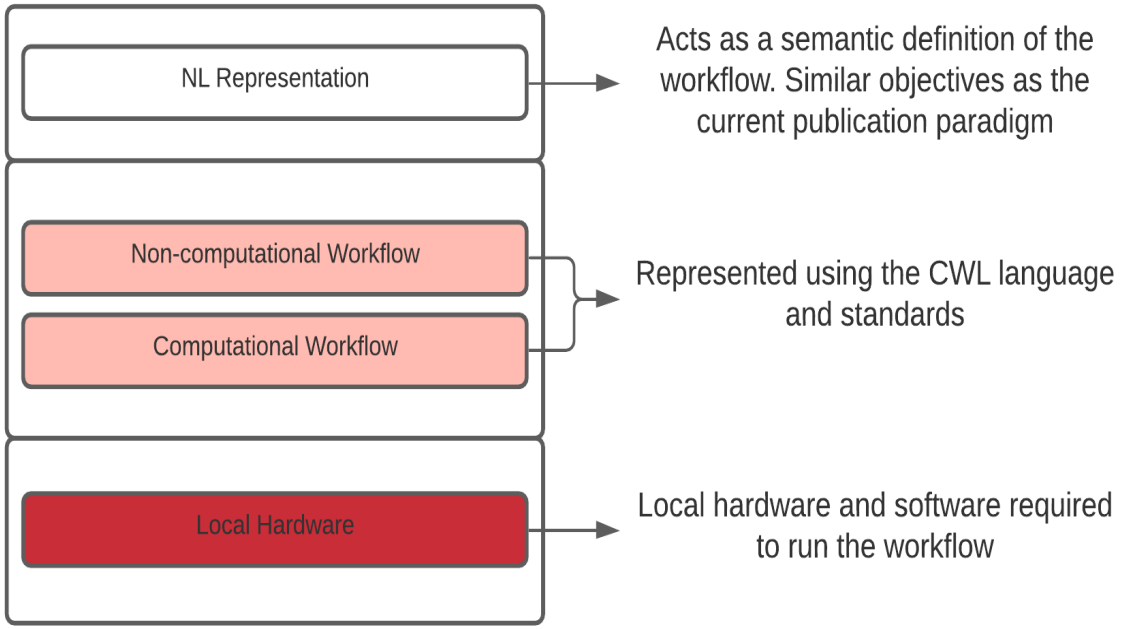
```
# Code based on the following notebook wit
# - pre-calculated benchmark.csv file is u
# - commented out code removed
# https://github.com/dmnfarrell/epitopepre

import os, sys, math

import numpy as np

import pandas as pd
pd.set_option('display.width', 130)

%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt

import seaborn as sns
sns.set_context("notebook", font_scale=1.4

import epitopepredict as ep
```

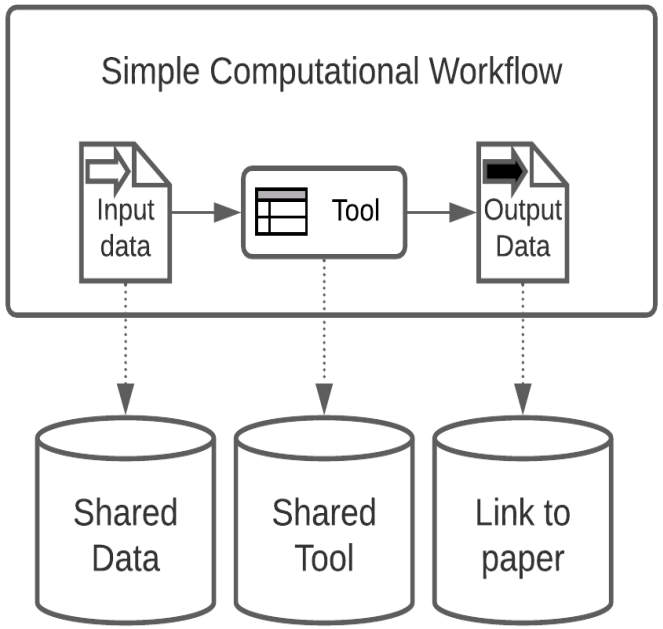



fish /home/nokome
nokome@venus ~ [64]> stencila -h
stencila 0.128.0
Stencila, in a terminal console, on your own machine
Enter interactive mode by using the `--interact` option with any command.
USAGE:
stencila [FLAGS] [OPTIONS] [SUBCOMMAND]
FLAGS:
-i, --interact Enter interactive mode (with any command and options as the prefix)
--debug Print debug level log events and additional diagnostics
-h, --help Prints help information
-V, --version Prints version information
OPTIONS:
--display <display>
Format to display results of commands (e.g. json, yaml, md)
--log-level <log-level>
The minimum log level to print [possible values: trace, debug, info, warn, error, never]
--log-format <log-format>
The format to print log events [possible values: simple, detail, json]
SUBCOMMANDS:
list List all open project and documents
open Open a project or document using a web browser
close Close a project or document
show Show a project or document
convert Convert a document to another format
diff Display the structural differences between two documents
merge Merge changes from two or more derived versions of a docu
with Run commands interactively with a particular project or do
projects Manage projects
documents Manage documents
sources Manage the current project's sources
codecs Manage codecs
parsers Manage parsers
kernels Manage kernels
config Manage configuration settings


fixtures
File Edit View Editor Window Help
README.md
articles
Makefile
code.md
coerce-1.yaml
coerce-2.yaml
elife-small.json
elife-small.json.media
era-plotly.json
era-plotly.json.media
kitchen-sink.ipynb
reshape-1.yaml
reshape-2.yaml
reshape-3.yaml
simple.Rmd
simple.docx
simple.html
simple.md
simple.tex
fragments
media
nodes
projects
daggy
README.md
code.md
1 This article fixture is focussed on the Markdown representation of executable code nodes such as `CodeChunk`, `CodeExpression`, and `Parameter` nodes.
2
3 ## Inline code
4
5 Code expressions have a language and the `exec` keyword in curly braces, like this `1+1`{r exec} and this `2+2`{python exec}. The language may be omitted e.g. `x`{exec} (in which case it may default to the file's language e.g. pymd files).
6
7 Double brace syntax is also supported, but generally not recommended e.g. {{2+3}}{python} an {{4+5}}.
8
9 Non-executable code fragments, lack the `exec` keyword but can have a language e.g. `3+3`{r}.
10
11 ## Block code
12
13 Code chunk use the `exec` keyword to differentiate them from code blocks,
14
15 ```r exec
16 "Hello from R"
17 ```
18
19 Non executable code blocks do not have the `exec` keyword,
20
21 ```python
22 # Not executed
23 ```
24
Markdown
This article fixture is focussed on the Markdown representation of executable code nodes such as CodeChunk, CodeExpression, and Parameter nodes.
Inline code
Code expressions have a language and the exec keyword in curly braces, like this and this . The language may be omitted e.g. (in which case it may default to the file's language e.g. pymd files).
Double brace syntax is also supported, but generally not recommended e.g. an .
Non-executable code fragments, lack the exec keyword but can have a language e.g. 3+3.
Block code
Code chunk use the exec keyword to differentiate them from code blocks,
1
"Hello from R"
R
No output to show

Packaging scientific workflows with Natural Language representations

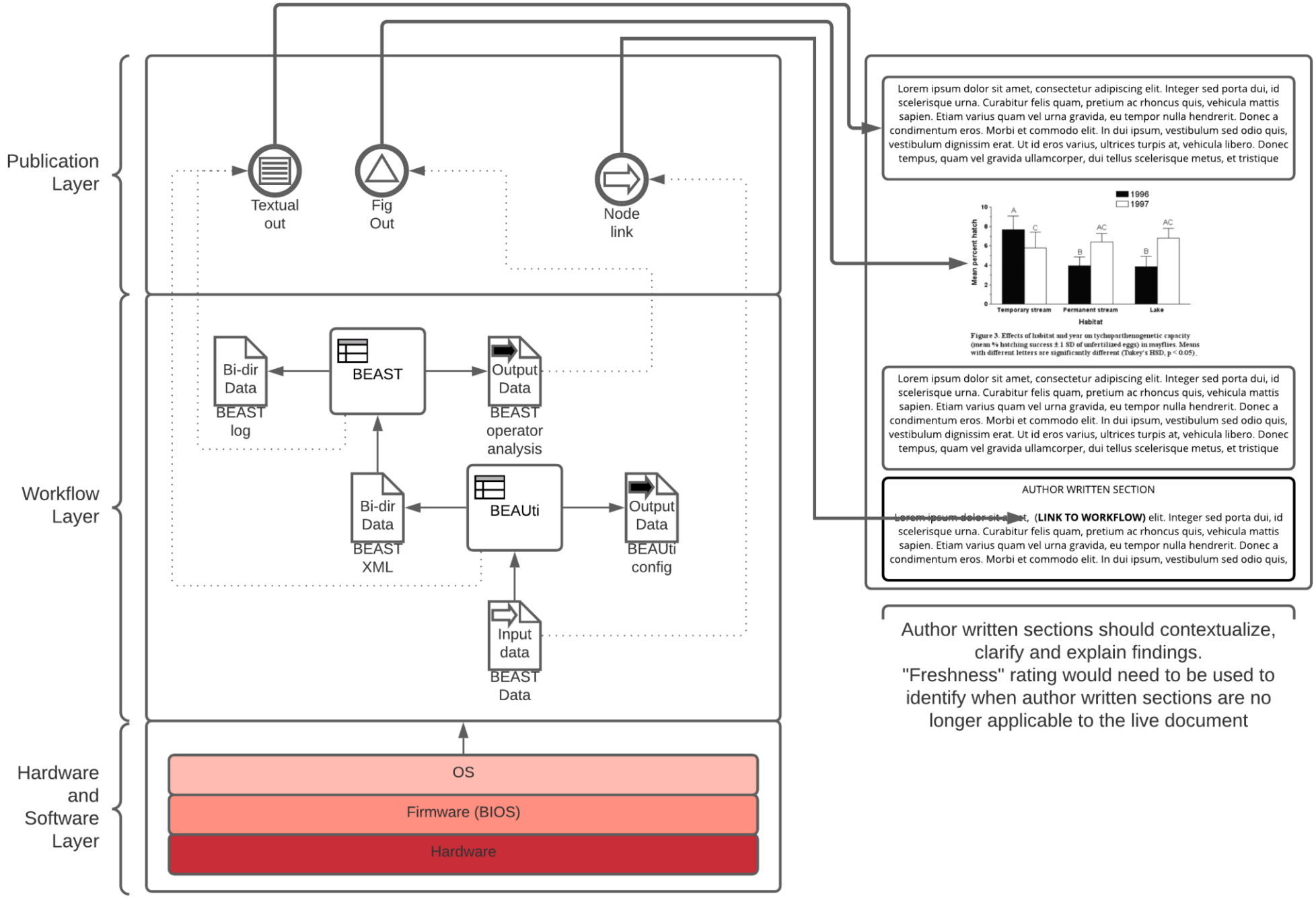




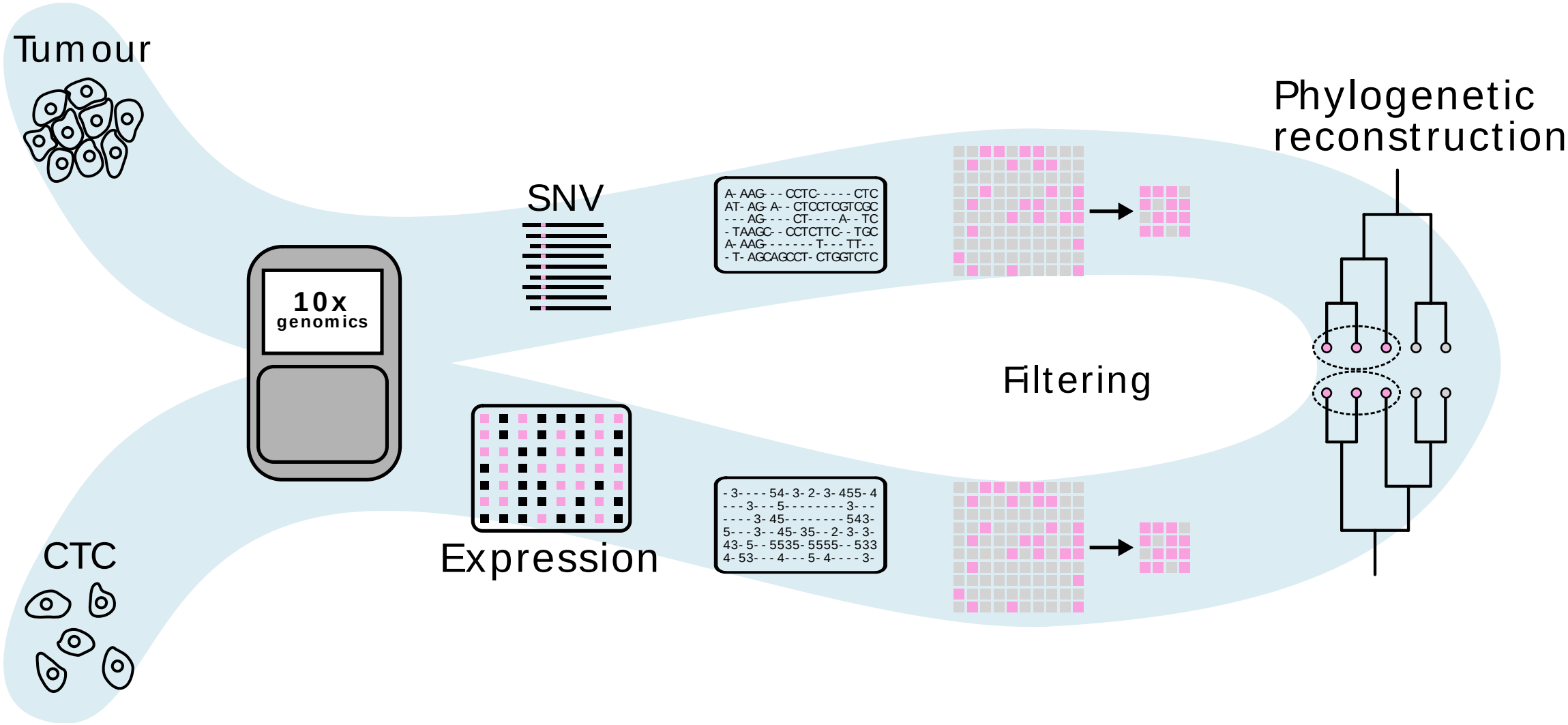
- Components of the computational workflow can be shared to repositories to enable reuse.
- Data flow, tools and components modelled in CWL (Common Workflow Language)
- When any change occurs, run the workflow and generate new outputs. These outputs are then instantiated within the NL Representation.

Goals:

- To provide a new format of publishing scientific articles that encapsulates the scientific workflow and enables live, up to date documentation.
- Enable reuse, reproducibility, transparency, trustworthy results, and easy deployment cross-platform.



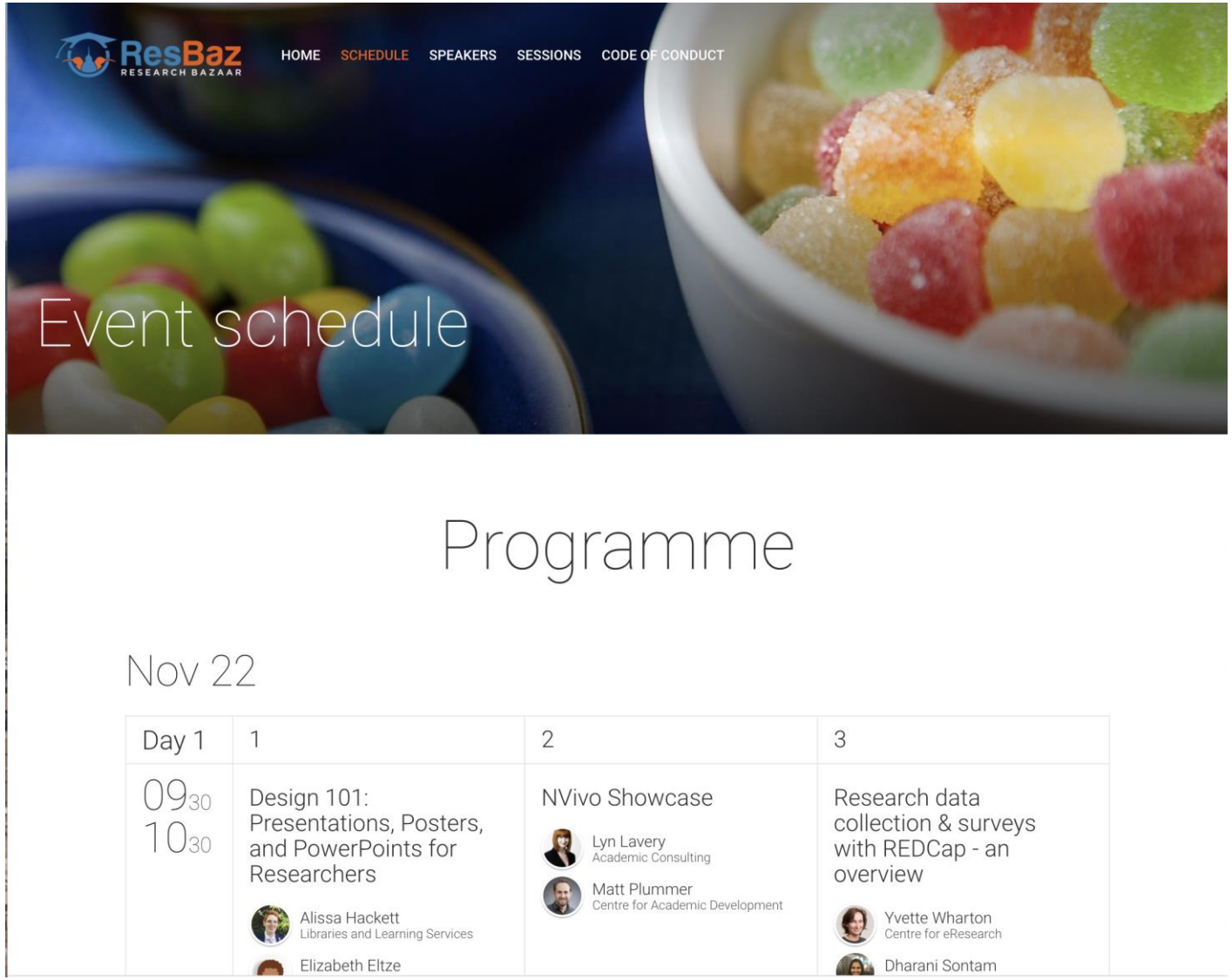
- BEAST computational workflow (Bioinformatics) underpins this example.
- If new input data is provided, the NL representation will reflect new workflow outcomes.
- Author written sections will need a fitness rating to determine if a section needs to be re-written upon new outcomes.


Tumour
CTC
10x
genomics
SNV
Expression
A- AAG- - - CCTC- - - - - CTC
AT- AG- A- - CTCCTCGTCGC
- - - AG- - - - CT- - - - A- - TC
- TAAGC- - CCTCTTC- - TGC
A- AAG- - - - - - - T- - - TT- - -
- T- AGCAGCCT- CTGGTCTC
- 3- - - - 54- 3- 2- 3- 455- 4
- - - 3- - - 5- - - - - - - - 3- - -
- - - - 3- 45- - - - - - - - 543-
5- - - 3- - 45- 35- - 2- 3- 3-
43- 5- - 5535- 5555- - 533
4- 53- - - 4- - - 5- 4- - - - 3-
Filtering
Phylogenetic
reconstruction

Some other highlights

- **Genomic epidemiology reveals transmission patterns and dynamics of SARS-CoV-2 in Aotearoa New Zealand**. *Nature Communications* 11, 6351. doi: 10.1038/s41467-020-20235-8 , cited 68 times already
- In the last year, our team has collectively produced around 50 journal and conference publications. Most of these publications are in top tier outlets.

Research Bazaar 2021, with Data Science Track



SESSION	REGISTRATION
Tidy data: an introduction	166
How to Make Your Publications Open Access	113
Introduction to R and RStudio	50
Resaerch computing with Rust programming language	???
Python for image manipulation and repeatable research pipelines	???
Infographics and storytelling	209
Tidyverse: key tips for existing R users	???
Introduction to R and RStudio additional session	50
Introduction to Julia	35
Machine learning in Julia	30
Machine Learning 101	156
How can Python help your research	205
Working with social media data?	93
Genomic data management: tips and tricks	51
Find, replace and manipulate big datasets	34
Introduction to geospatial tools and manipulations in R	40
Introduction to OpenRefine	???
Data analysis with Jupyter Notebooks	???
Introduction to High Performance Computing with NeSI	18
Version control for documents and code	39
An introduction to processing remote sensing data with Google Earth Engine	29
Becoming a Data Scientist	OPEN
Open Source in Research	42
High performance computations with multithreading	23
Parallel programming with MPI	27
TOTAL	1410