BHK LAB

Improving Reproducibility in Machine Learning Research

Benjamin Haibe-Kains

bhaibeka@uhnresearch.ca

@bhaibeka

Senior Scientist, Princess Margaret Cancer Centre
Associate Professor, University of Toronto
Affiliate, OICR and Vector Institute

May 21, 2021

Abstract

While biostatistics and machine learning are essential to analyze biomedical data, researchers are facing multiple challenges around research reproducibility and transparency. Given the proliferation of studies investigating the applications of biostatistics and machine learning in research and clinical studies, it is essential for independent researchers to be able to scrutinize and reproduce the results of a study using its materials, and build upon them in future studies.

Computational reproducibility is achievable when the data can easily be shared and the required computational resources are relatively common. However, the complexity of current algorithms and their implementation, the need for specific computer hardware and the use of sensitive biomedical data represent major obstacles in healthy-related research. In this talk, I will describe the various aspects of a typical biomedical study that are necessary for reproducibility and the platforms that exist for sharing these materials with the scientific community.

Disclosures

- I am a co-Founder of the MAQC (Massive Analysis and Quality Control) Society
- I am part of the Scientific Advisory Board of:
  - Consortium de recherche biopharmaceutique (CQDM), Quebec, Canada
  - Break Through Cancer, Commonwealth Cancer Consortium, United States
  - Canadian Institute of Health Research - Institute of Genetics, Canada
- I am part of the Executive Committee of the Terry Fox Digital Health and Discovery Platform, Canada
- I am part of the Board of Directors of AACR International - Canada, The American Association for Cancer Research, United States
- I am a consultant for Code Ocean Inc, United States

The Radiomics Heroes @ BHKLAB

Caryn Geady

Mattea Welch

Sejin Kim

Jun Won Kim

Joseph Marsilla

Michal Kazmierski

Reza Reiazi

BHK LAB

What is reproducibility?

Replication

Scientific claims are confirmed by completely independent investigations

Reproducibility

Ability of independent investigators to re-create the results claimed by the original investigators using the original samples, assays, data and analysis techniques

Computational Reproducibility

Running the same computer code on the same data to obtain the same results

arXiv.org

Reproducible Research: A Retrospective

Roger D. Peng, Stephanie C. Hicks *[Submitted on 23 Jul 2020]*

Why does reproducibility matter?

- As data analysis pipelines become increasingly complex, a high level of **transparency** is required to allow the scientific community to properly scrutinize a study

- An analysis may contain many (hyper)parameters, understanding their impact on results is necessary to assess the **robustness** of the claims

- Sharing all study materials maximize the **reusability** of the research outputs (higher impact and return on investment)

The dark side of AI

nature

International evaluation of an AI system for breast cancer screening

https://doi.org/10.1038/s41586-019-1799-6
Received: 27 July 2019
Accepted: 5 November 2019
Published online: 1 January 2020

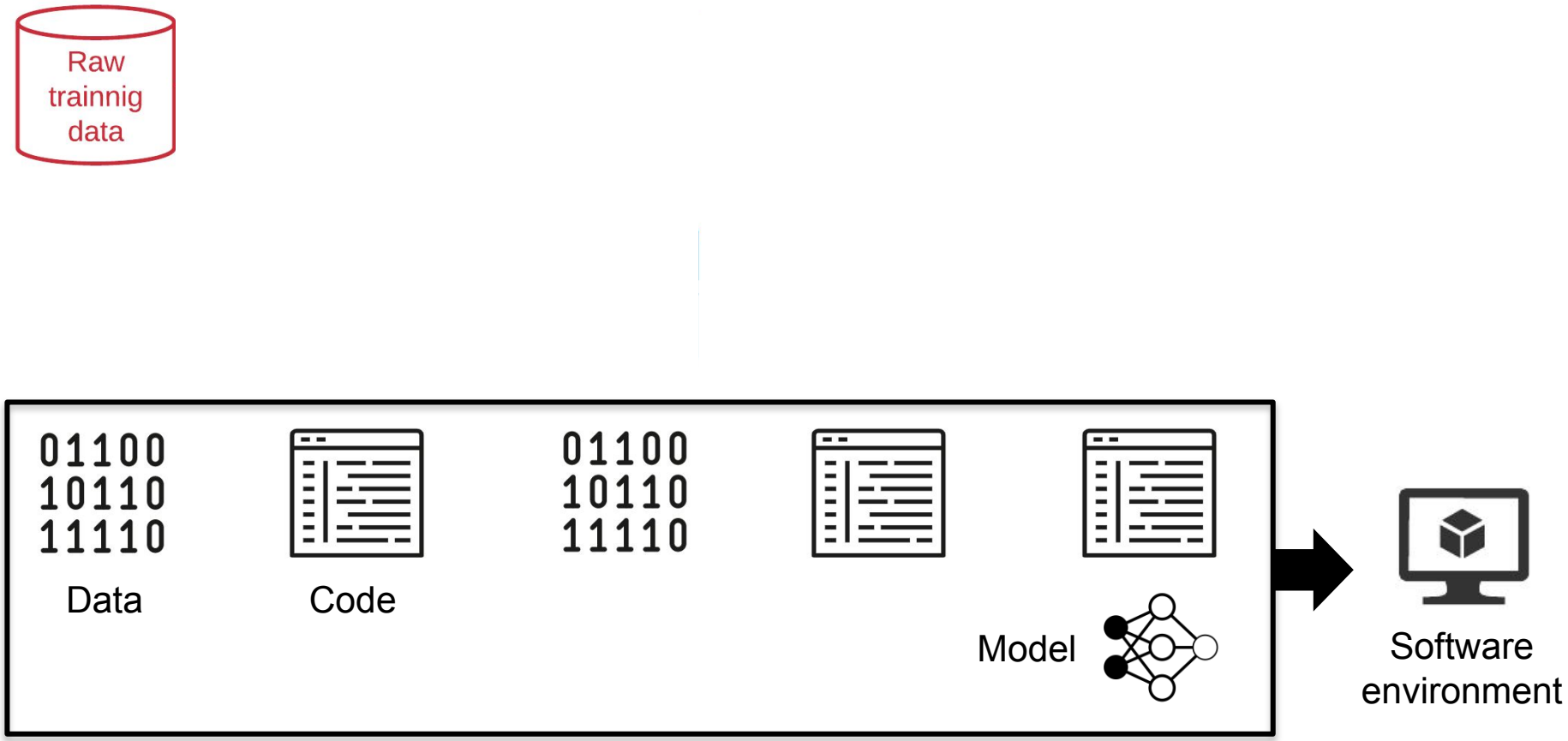
Scott Mayer McKinney[1,14*], Marcin Sieniek[1,14], Varun Godbole[1,14], Jonathan Godwin[2,14], Natasha Antropova[2], Hutan Ashrafian[3,4], Trevor Back[2], Mary Chesus[2], Greg C. Corrado[1], Ara Darzi[3,4,5], Mozziyar Etemadi[6], Florencia Garcia-Vicente[6], Fiona J. Gilbert[7], Mark Halling-Brown[8], Demis Hassabis[2], Sunny Jansen[9], Alan Karthikesalingam[10], Christopher J. Kelly[10], Dominic King[10], Joseph R. Ledsam[2], David Melnick[6], Hormuz Mostofi[1], Lily Peng[1], Joshua Jay Reicher[11], Bernardino Romera-Paredes[2], Richard Sidebottom[12,13], Mustafa Suleyman[2], Daniel Tse[1*], Kenneth C. Young[8], Jeffrey De Fauw[2,15] & Shravya Shetty[1,15*]

[1]Google Health, Palo Alto, CA, USA. [2]DeepMind, London, UK. [3]Department of Surgery and Cancer, Imperial College London, London, UK. [4]Institute of Global Health Innovation, Imperial College London, London, UK. [5]Cancer Research UK Imperial Centre, Imperial College London, London, UK. [6]Northwestern Medicine, Chicago, IL, USA. [7]Department of Radiology, Cambridge Biomedical Research Centre, University of Cambridge, Cambridge, UK. [8]Royal Surrey County Hospital, Guildford, UK. [9]Verily Life Sciences, South San Francisco, CA, USA. [10]Google Health, London, UK. [11]Stanford Health Care and Palo Alto Veterans Affairs, Palo Alto, CA, USA. [12]The Royal Marsden Hospital, London, UK. [13]Thirlestaine Breast Centre, Cheltenham, UK. [14]These authors contributed equally: Scott Mayer McKinney, Marcin T. Sieniek, Varun Godbole, Jonathan Godwin. [15]These authors jointly supervised this work: Jeffrey De Fauw, Shravya Shetty. *e-mail: scottmayer@google.com; tsed@google.com; sshetty@google.com

Code availability

The code used for training the models has a large number of dependencies on internal tooling, infrastructure and hardware, and its release is therefore not feasible. However, all experiments and implementation details are described in sufficient detail in the Supplementary Methods section to support replication with non-proprietary libraries. Several major components of our work are available in open source repositories: Tensorflow (https://www.tensorflow.org); Tensorflow Object Detection API (https://github.com/tensorflow/models/tree/master/research/object_detection).

Key elements of a typical AI pipeline



Sharing raw and processed data

- For data that are not sensitive and can be released publicly
  - Large files: established repositories specific to each field/domain, such as the European Genotype-phenotype Archive (*EGA*) for genomics
  - Small files: immutable data repositories assigning unique identifiers such as *Zenodo*, *Harvard Dataverse*, *Dryad*, or *figshare*

- For data that are private and/or sensitive (eg, patient health information)
  - Set up a Data Access Committee (DAC) and a transparent process to review requests (with timeline)
  - Use established repositories that can forward requests to DAC and grant access upon approval, such as *EGA* or *dbGaP*
  - Ensure that the data transfer is secure and automated (manual transfer of data on hard disk and shipping by mail is not ideal...)

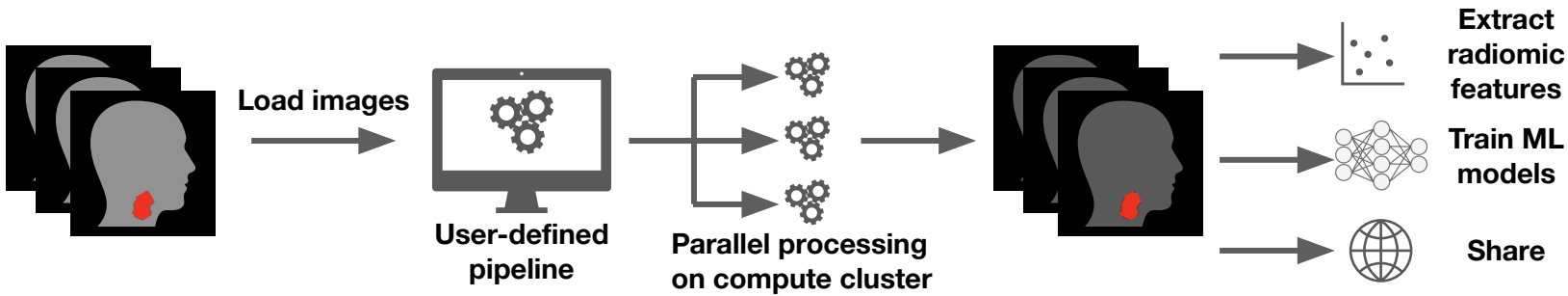
Raw data ≠ analysis-ready

BHK LAB

In the lab: Imgtools for processing large imaging datasets

github.com/bhklab/imgtools

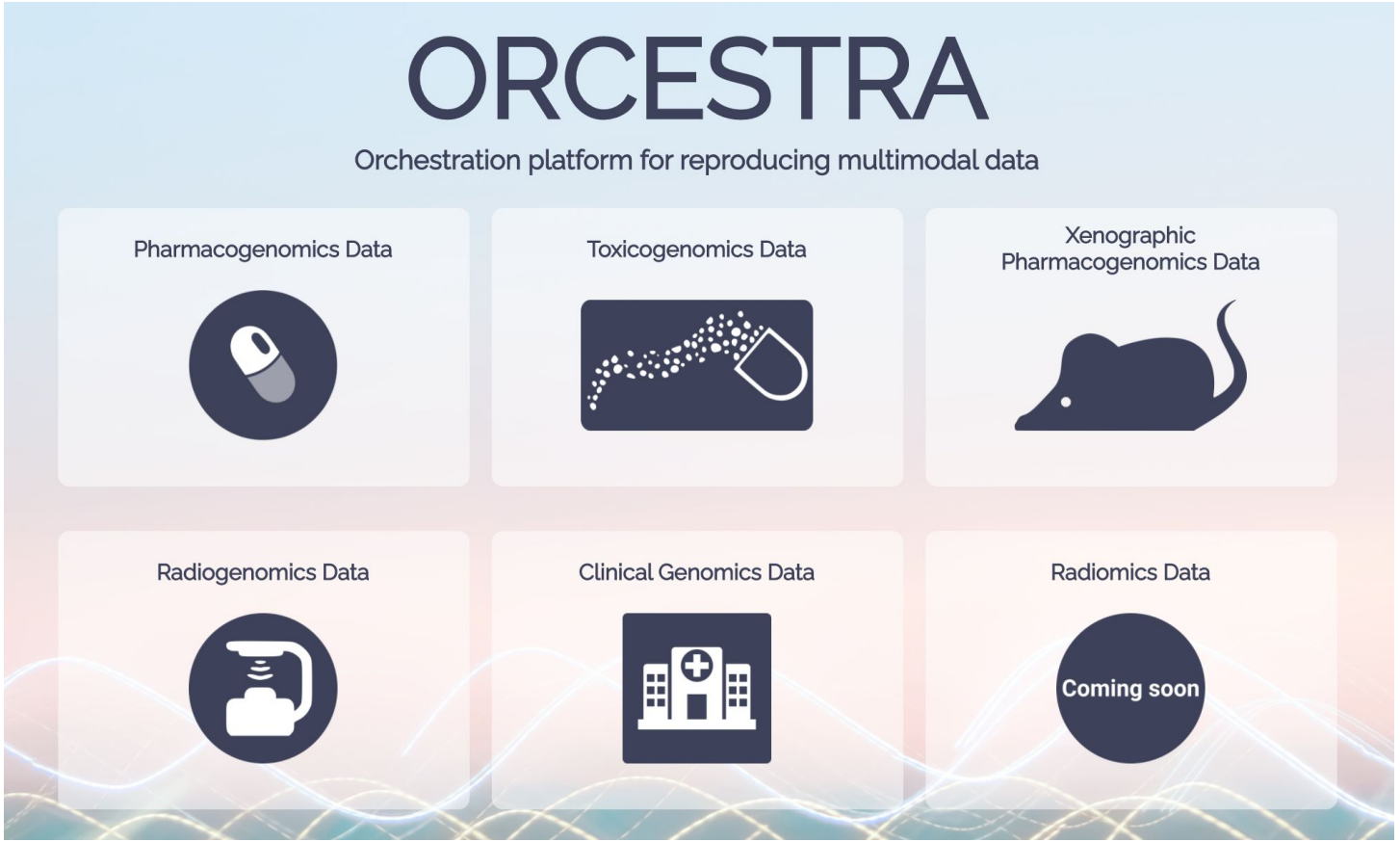
- Built in Python using tested open-source components
- Rich set of standardised image processing operations
- Easy sharing of pipeline specifications
- Support for DICOM (including RT Structure Set)



BHK LAB

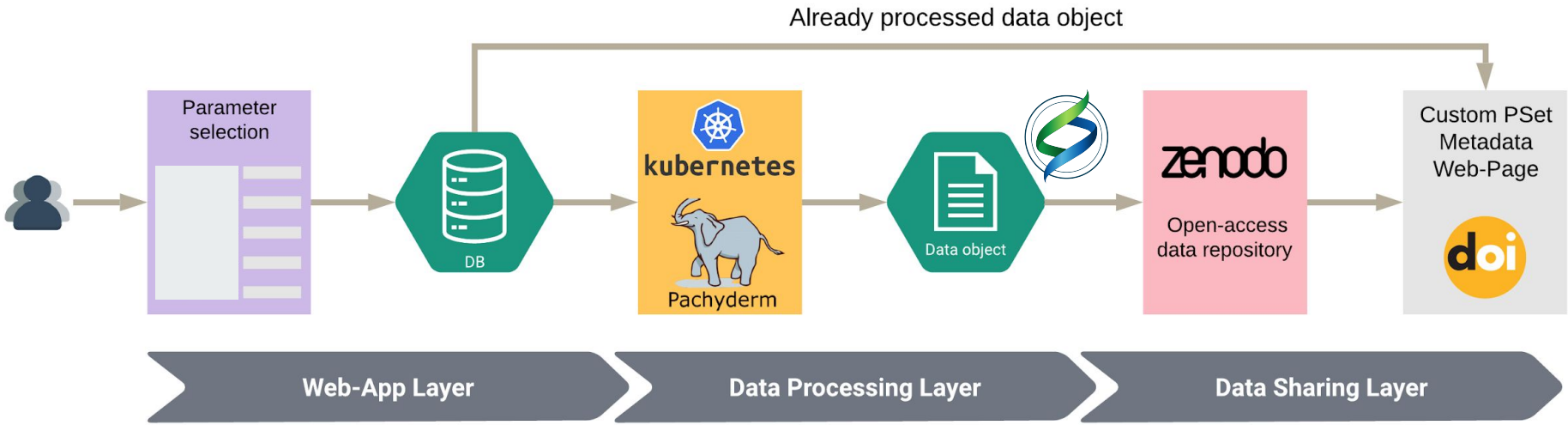
In the lab: Automating data processing and versioning

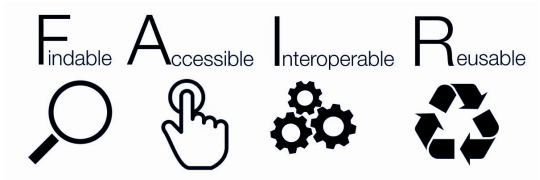
orcestra.ca



BHK LAB

In the lab: Automating data processing and versioning

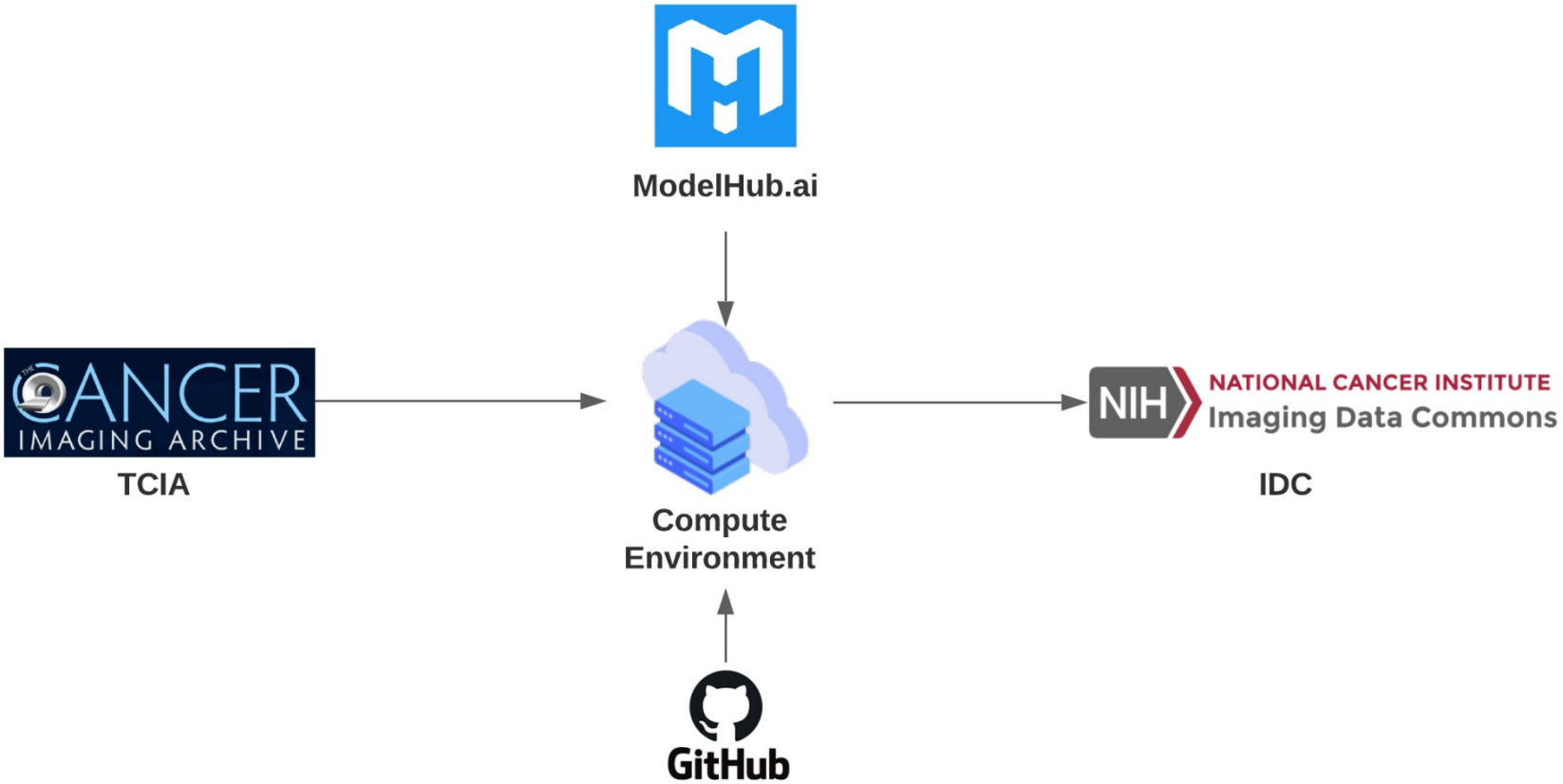




➔ Maximize cross-dataset operability by standardizing processing pipelines and annotations (meta and clinical data)

BHK LAB

For the community



Sharing raw and processed data

For data that are private and/or sensitive (eg, patient health information)

- Set up a Data Access Committee (DAC) and a transparent process to review requests (with timeline)
- Use established repositories that can forward requests to DAC and grant access upon approval, such as *EGA* or *dbGaP* for genomics
- Ensure that the data transfer is secure and automated (manual transfer of data on hard disk and shipping by mail is not ideal...)

TCIA is using a different approach

- Most studies are open, few are protected (eg, head and neck due to potential face reconstruction)
- Requesters signed a Data Usage Agreement (DUA) with TCIA
- TCIA takes on the liability
- Identity of requesters can be communicated to the data generators

Sharing computer code

- Three levels of computer code:
  - Preprocessing of the raw data, including curation of metadata, quality control and normalization
  - Training of the predictive model (fitting of the model, tuning, testing)
  - Validation of the final (fully specified) predictive model on independent data
- Code can easily be shared on established versioning systems, like *GitHub* or *BitBucket*
- Multiple frameworks exists for deep learning: *PyTorch*, *TensorFlow*
- Higher level frameworks like *Keras* and *PyTorch Lightning* allow for better organization of the code, making it easier to read
- Documentation is a must!
- Think about your license: open-source? Non-commercial?

Sharing predictive models

- Final models can also be shared to facilitate application on new data
- Platforms exist to easily share models, their architecture and full set of parameters
  - *ModelHub*, *ModelDepot*, *Model Zoo*
- An example of code goes a long way to showcase how the final model can be used
  - Describe how to format/process the input data, how to interpret the outputs (predictions), and the limitations of the model
- A model may reveal key aspects of your dataset, and be used to reveal the identity of some participants
  - Differential privacy can sometimes be ensured
  - Otherwise share your model via a data access committee

The power of virtualization

- Having access to data and code is not sufficient to reproduce the results of a study
- You also need to set up the software environment
  - Operating system, software libraries (versions?)
  - Even the type of hardware matters (eg, numerical precision)

- Go **virtual**! (I mean *virtual machines* and *containers*…)
  - Specify all the required libraries and their versions
  - Once the container is built, add your code and data
  - Complete a successful run, check your results and… voilà!

- Users can now easily **reproduce** all your results, **scrutinize** and **challenge** them
- Many platforms exist to help build and share containers in the cloud: *Code Ocean, Gigantum, Docker Hub*, ...

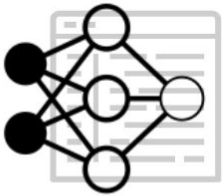
Levels of reproducibility - Level 1

Raw data

Code for QC and preprocessing

Processed data

Code for model training



Final model and code

Software environment

- All elements are shared, the study is fully reproducible
- Providing a container will **really** help the community build upon your study

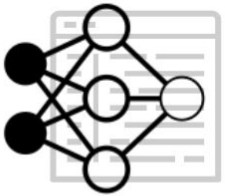
Levels of reproducibility - Level 2

Raw data

Code for QC and preprocessing

Processed data

Code for model training



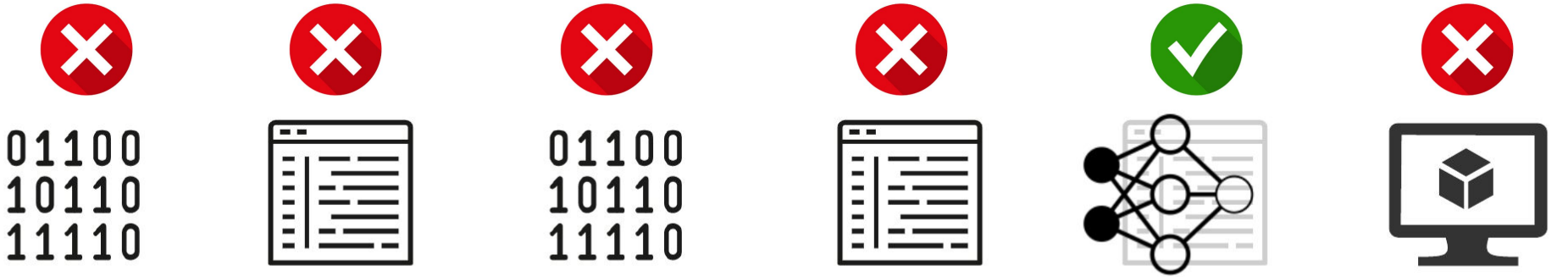
Final model and code

Software environment

- Raw data could not be shared but processed data available
- If the raw data and processing are well documented, it is *OK*
- But new processing approaches or meta-analyses might not be feasible

BHK LAB

Levels of reproducibility - Level 3



- The study is NOT reproducible
- But the application of the model in new datasets can still be explored

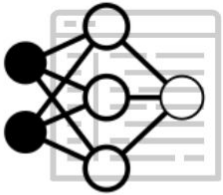
Levels of reproducibility - Level UNDEF

Raw data

Code for QC and preprocessing

Processed data

Code for model training



Final model and code

Software environment

- Advertisement for a cool but closed technology

Initial model vs derivatives

- As machine learning models become more complex (eg, deep neural networks), more computational resources are required
  - Large amounts of CPUs, GPUs or memory
  - Specific hardware that are expensive and/or hard to find (even proprietary technologies)

- It is only a matter of time until the technology becomes cheaper and more accessible

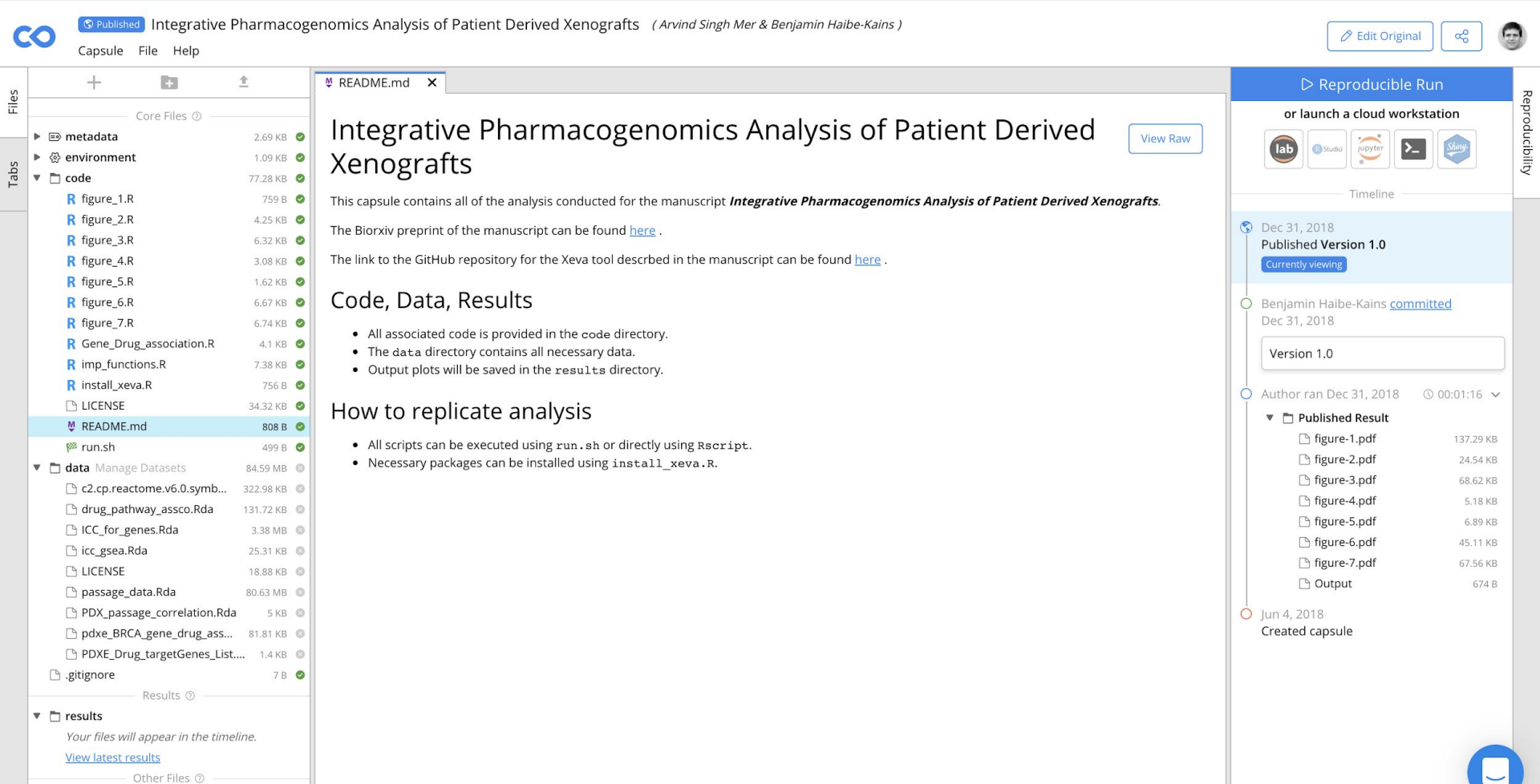
- Meantime, further research can be done in **simplifying** the initial model
  - Decrease the dependency on specific hardware
  - Clever updates in how the model is trained or its architecture can go a long way
  - Broaden the user base, increase impact

Research Reproducibility in Practice

Tools to improve reproducibility - Code Ocean

- Sharing data
  - Public data: Zenodo, Harvard Dataverse, Dryad, figshare, ...
  - Private/sensitive data: dbGaP, NCBI SRA, EBI EGA, ...
- Sharing computer code
  - GitHub, GitLab, Bitbucket, ...
- Sharing software environment
  - **Code Ocean**, Gigantum, Docker Hub, ...
- Sharing computational models
  - ModelHub, ModelDepot, Model Zoo, ...
- Sharing analysis results
  - Zenodo, Harvard Dataverse, Dryad, figshare, ...

Code Ocean interface



How do we use Code Ocean in my lab?

- **Over the course of the analysis**
  - Implementation of a capsule to run all the analyses from the beginning
  - Good tool for collaboration as anyone is using the same software environment
- **Internal review prior to the manuscript submission**
  - Independent review of the code and rerun of all the analyses from the raw or low-level processed data
- **At the time of the manuscript submission**
  - Strengthen the study, show the reviewers that the research is transparent and increase its impact (reusability)
- **After publication**
  - For older publications or collaborative papers with time-sensitive submission
  - A great way to update the results when new data are released, improved methods are available or bugs are found

Continuous development

Using **Code Ocean** during the implementation of an analysis pipelines has multiple benefits:

- Quickly install software packages
- A well-controlled software environment facilitates the debugging process
  - You can literally share the capsule with a colleague to get help
- Optimize collaboration as all parties use the exact same data and software environment
  - Co-editing of code can be challenging
- Running code many times for large analyses may require large computational resources

  → ***Co-Exist*** (beta)
- It ensures full reproducibility by construction

Pre-publication review

The analysis is done, results looking good!

- Time to clean and document the code → **Build a Code Ocean capsule!**
- Paper ready to submit but … let's have an independent review first
- Another lab member, not involved or only peripherally involved in the study checks the code
- And rerun everything from the raw data or low-level processed data

- **Surprising how often the results change!**

Post-publication review

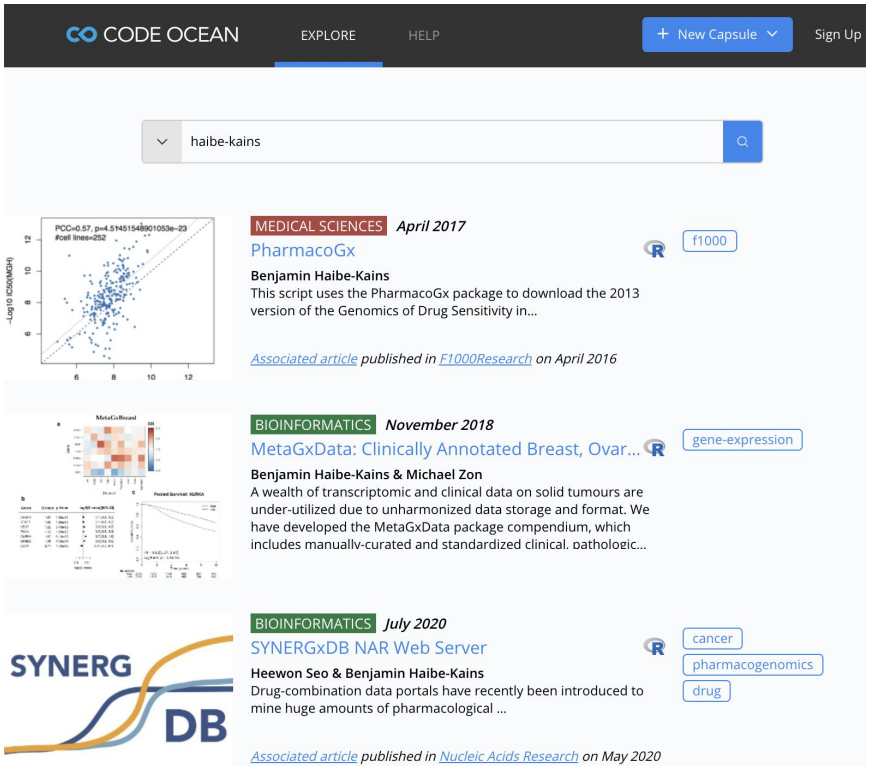
- Your analysis yields unexpected results (expect a fair share of scrutiny)

- You work hard on making your study fully reproducible and prepare the data and computer code for public release

- Your paper is finally accepted! Let the fun begin ...

Adoption in the lab

- 16 publications with a Code Ocean capsule since 2017
- A capsule is mandatory for **all** new studies led by the lab



Challenges:

- Private data?
- Proprietary software?
- Need for expensive computational resources?

BHK LAB

Platforms we used in the lab

- **R** or **Python** for the analyses
  - R packages on **CRAN / Bioconductor**
  - Python packages on **Pypi / Anaconda**
- **PyTorch Lightning** for deep learning
- **GitHub** for code versioning
- **Code Ocean** to share code and software environment
- **Zenodo** for sharing specific research outputs (processed data, results,…)
- European Genotype-phenotype Archive (**EGA**) and The Cancer Imaging Archive (**TCIA**) for sharing large genomics and imaging data, respectively
- **ORCESTRA** to process, version and share the data *
- **ModelHub** to share deep neural networks

Lessons learned

- **Making your research fully reproducible takes time and expertise**
  - Include these costs in the budget of your grant proposals

- **Put research reproducibility at the center of your lab's modus operandi and mission**
  - Discuss research reproducibility during interviews
  - Set expectations for new recruits
  - Make it mandatory, no leeway (exceptions must be strongly justified)

- **Resist the pressure!**
  - We are all in a hurry to publish, competition is tough …
  - Editors and Reviewers appreciate transparency
  - It is an investment, addressing the reviewer's comments and building upon previous results will be greatly facilitated

Lessons learned

- Knowing that data/code and software environment will be shared is a **strong motivation to improve the quality of the research itself**
  - Containers facilitate the pre- and post-publication review
- Containers make it easier to **collaborate**
  - All participants use exactly the same software environment
  - Easier to check whether the code runs properly after an update
- Sharing research outputs **increases impact**, opportunities for new collaborations
- Open Science and research reproducibility **do not prevent commercialization**

PNAS February 10, 2015 112 (6) 1645-1646

Opinion: Reproducible research can still be wrong: Adopting a prevention approach

Jeffrey T. Leek and Roger D. Peng

BHK LAB

Matters arising

nature

Transparency and reproducibility in artificial intelligence

Benjamin Haibe-Kains[1,2,3,4,5]✉, George Alexandru Adam[3,5], Ahmed Hosny[6,7], Farnoosh Khodakarami[1,2], Massive Analysis Quality Control (MAQC) Society Board of Directors*, Levi Waldron[8], Bo Wang[2,3,5,9,10], Chris McIntosh[2,5,9], Anna Goldenberg[3,5,11,12], Anshul Kundaje[13,14], Casey S. Greene[15,16], Tamara Broderick[17], Michael M. Hoffman[1,2,3,5], Jeffrey T. Leek[18], Keegan Korthauer[19,20], Wolfgang Huber[21], Alvis Brazma[22], Joelle Pineau[23,24], Robert Tibshirani[25,26], Trevor Hastie[25,26], John P. A. Ioannidis[25,26,27,28,29], John Quackenbush[30,31,32] & Hugo J. W. L. Aerts[6,7,33,34]

Published online: 14 October 2020

Acknowledgements

Andrew Hope

Scott Bratman

Tom Purdie

Chris Mcintosh

Alejandro Berlin

Michael Milosevic

Fei-Fei Liu

Hugo Aerts

Benjamin Kann

BHK LAB

The Gattuso Slaight Personalized Cancer Medicine Fund

The Princess Margaret Cancer Foundation UHN

CIHR IRSC

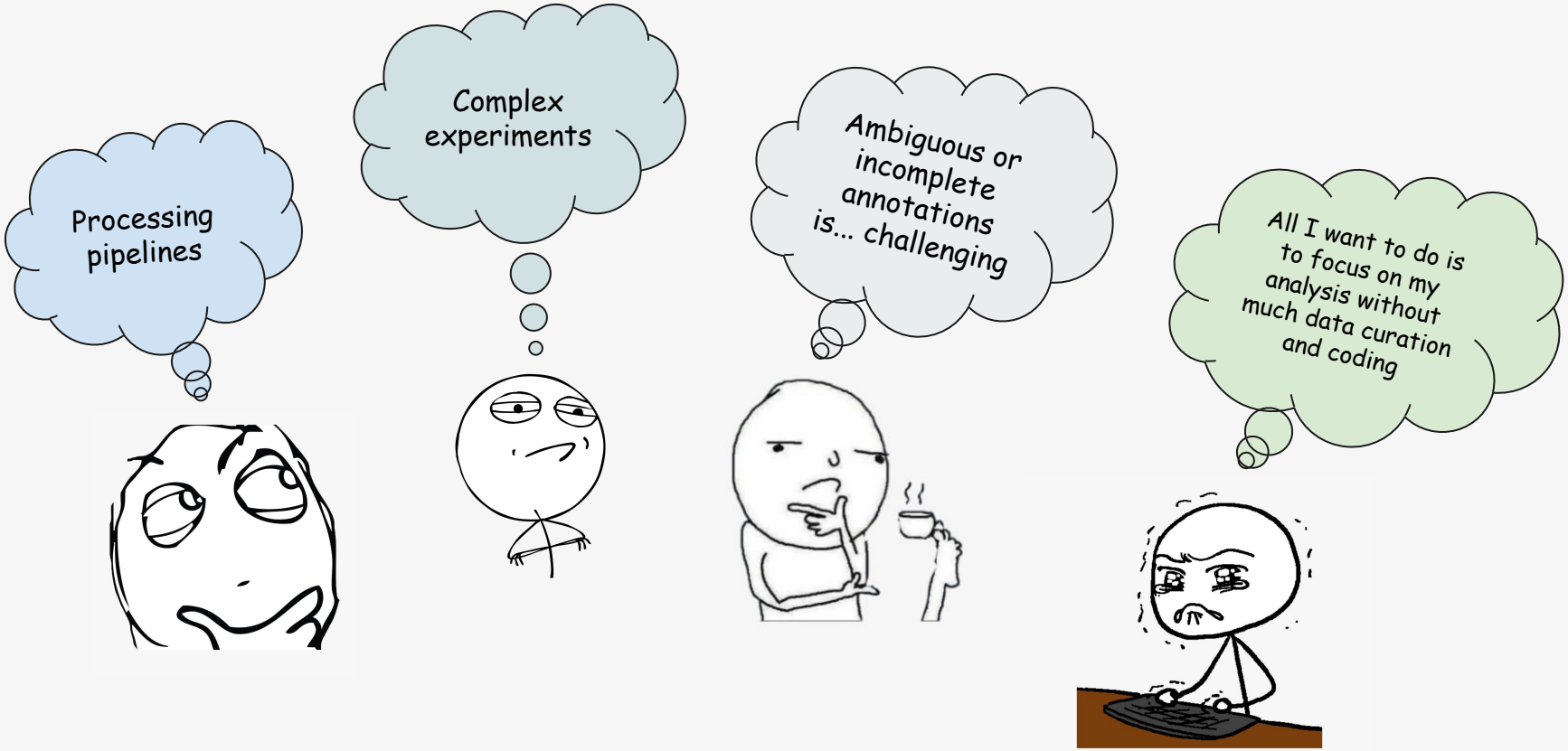
NSERC CRSNG

Thank you for your attention

Thank you for your attention

APPENDIX

Challenges


Processing pipelines
Complex experiments
Ambiguous or incomplete annotations is... challenging
All I want to do is to focus on my analysis without much data curation and coding

The reality of many biomedical labs

Today

- No coding expertise, use of tools like Excel and PRISM to analyze the data
  - Prone to errors, untraceable
  - **Not reproducible**
- Lack of data coordination
- No software development or difficult to maintain
- Journal guidelines typically not properly enforced
  - Easy to get away from sharing data or code

Tomorrow?

- Funding agencies support data coordination and research reproducibility
- Journals require and assess research reproducibility
- Use of open programming languages to develop analysis pipelines
  - Documentation
  - Automated/reproducible
- Data are annotated and shared internally and externally
- Data and code grow/improve over time

How do we use Code Ocean in my lab?

- **Over the course of the analysis**
  - Implementation of a capsule to run all the analyses from the beginning
  - Good tool for collaboration as anyone is using the same software environment
- **Internal review prior to the manuscript submission**
  - Independent review of the code and rerun of all the analyses from the raw or low-level processed data
- **At the time of the manuscript submission**
  - Strengthen the study, show the reviewers that the research is transparent and increase its impact (reusability)
- **After publication**
  - For older publications or collaborative papers with time-sensitive submission
  - A great way to update the results when new data are released, improved methods are available or bugs are found

Continuous development

Using **Code Ocean** during the implementation of an analysis pipelines has multiple benefits:

- Quickly install software packages
- A well-controlled software environment facilitates the debugging process
  - You can literally share the capsule with a colleague to get help
- Optimize collaboration as all parties use the exact same data and software environment
  - Co-editing of code can be challenging
- Running code many times for large analyses may require large computational resources

  → ***Co-Exist*** (beta)
- It ensures full reproducibility by construction

Pre-publication review

The analysis is done, results looking good!

- Time to clean and document the code → **Build a Code Ocean capsule!**
- Paper ready to submit but … let's have an independent review first
- Another lab member, not involved or only peripherally involved in the study checks the code
- And rerun everything from the raw data or low-level processed data

- **Surprising how often the results change!**

Post-publication review

- Your analysis yields unexpected results (expect a fair share of scrutiny)

- You work hard on making your study fully reproducible and prepare the data and computer code for public release

- Your paper is finally accepted! Let the fun begin ...

Genomics of Drug Sensitivity in Cancer (GDSC)

Cancer Genome Project

Wellcome Trust Sanger Institute/Massachusetts General Hospital

727 cancer cell lines

Affymetrix HG-U133A 12,153 genes

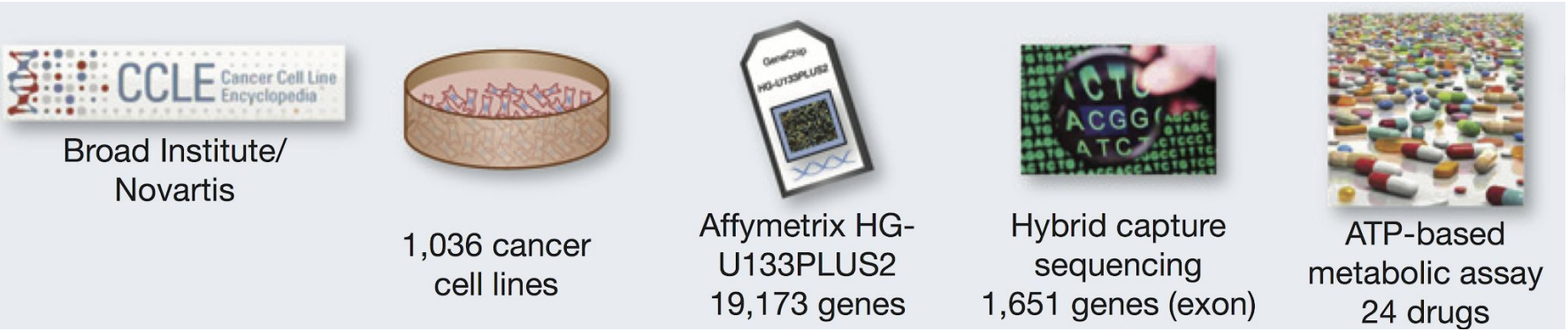
Sanger capillary sequencing 68 genes (exon)

Reductase-activity metabolic assay 138 drugs

"By linking drug activity to the functional complexity of cancer genomes, systematic pharmacogenomic profiling in cancer cell lines provides a powerful biomarker discovery platform to guide rational cancer therapeutic strategies"

Garnett et al., Nature 2012

Cancer Cell Line Encyclopedia (CCLE)



"The generation of genetic predictions of drug response in the preclinical setting and their incorporation into cancer clinical trial design could speed the emergence of 'personalized' therapeutic regimens."

Barretina et al., Nature 2012

Comparing large pharmacogenomic studies

PharmacoGx

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

nature

ANALYSIS RESEARCH

Inconsistency in large pharmacogenomic studies

Benjamin Haibe-Kains[1,2], Nehme El-Hachem[1], Nicolai Juul Birkbak[3], Andrew C. Jin[4], Andrew H. Beck[4]*, Hugo J. W. L. Aerts[5,6,7]* & John Quackenbush[5,8]*

Published: 27 November 2013

F1000Research
Open for Science

RESEARCH ARTICLE

Revisiting inconsistency in large pharmacogenomic studies

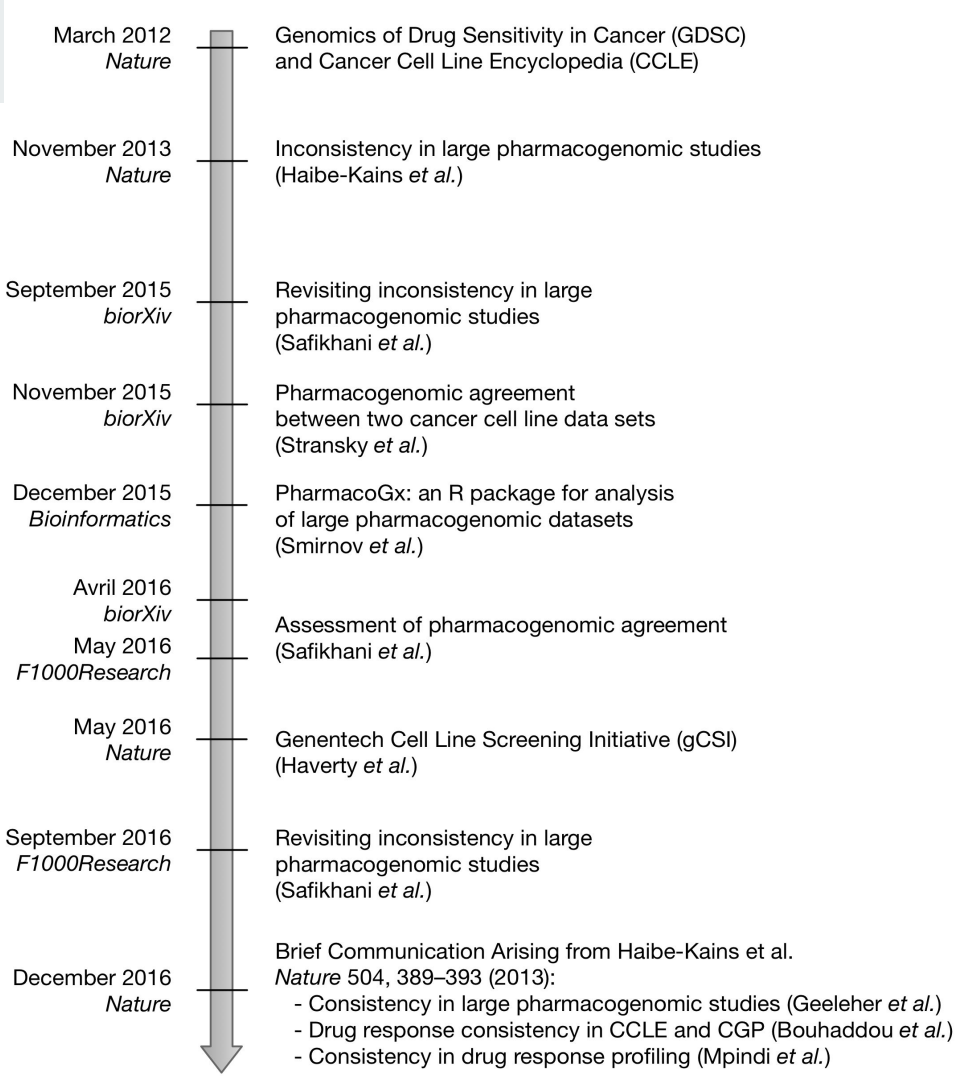
Open in Code Ocean

Zhaleh Safikhani[1,2], Petr Smirnov[1], Mark Freeman[1], Nehme El-Hachem[3], Adrian She[1], Quevedo Rene[1,2], Anna Goldenberg[4,5], Nicolai J. Birkbak[6], Christos Hatzis[7,8], Leming Shi[9,10], Andrew H. Beck[11], Hugo J.W.L. Aerts[13,14], John Quackenbush[12,14], Benjamin Haibe-Kains[1,2,5,15]

Version 1 16 Sep 16

An epic controversy

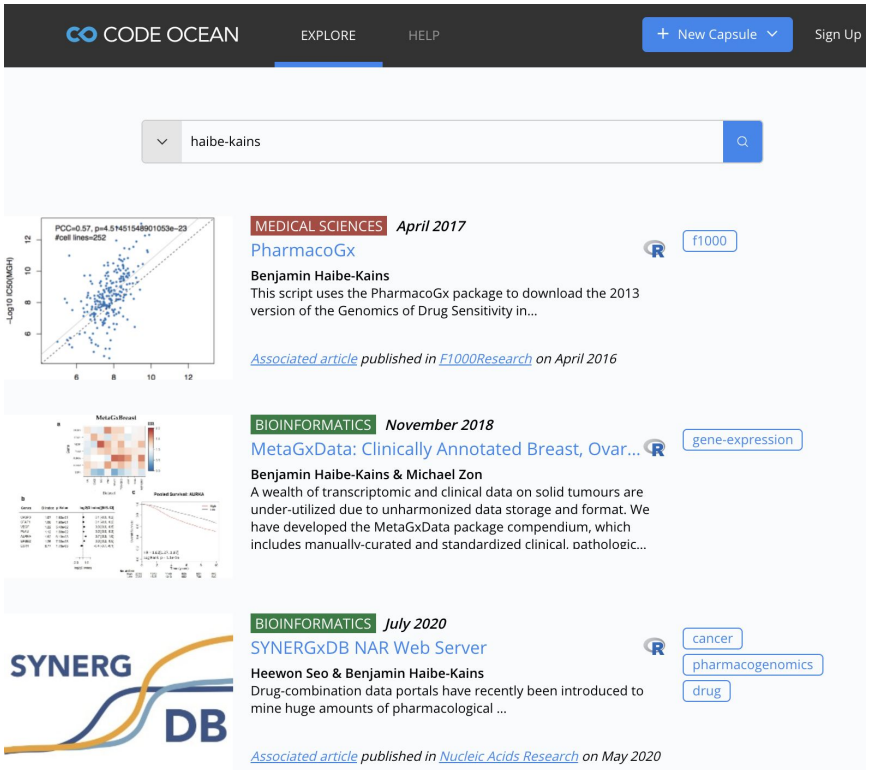
- Transparency (sharing of curated data, computer code and all results) helped us address critics
- Reproducibility and high-quality software helped us update, adapt and improve the analysis quickly
- The shared data and code triggered new research in the field



BHK LAB

Adoption in the lab

- 16 publications with a Code Ocean capsule since 2017
- A capsule is mandatory for **all** new studies led by the lab



Challenges:

- Private data?
- Proprietary software?
- Need for expensive computational resources?

BHK LAB

Back to your own lab

1. **Restrict the use of Excel** and similar tools to *data visualization*
    - A computer program (code) is a more efficient way to transform and analyze data
2. **Nominate a lab member** to look into the various platforms to increase transparency and reproducibility. For instance:
    - R or Python for the analyses
    - GitHub for code versioning
    - Code Ocean to share code and software environment
    - Zenodo for sharing specific research outputs (processed data, results, …)
3. **Implement a test run** with a published study
    - Can you reproduce the results?
    - Share the data, code, capsule, and results, enjoy the increased visibility/impact
4. **Build the expertise**, try with a new study to be submitted

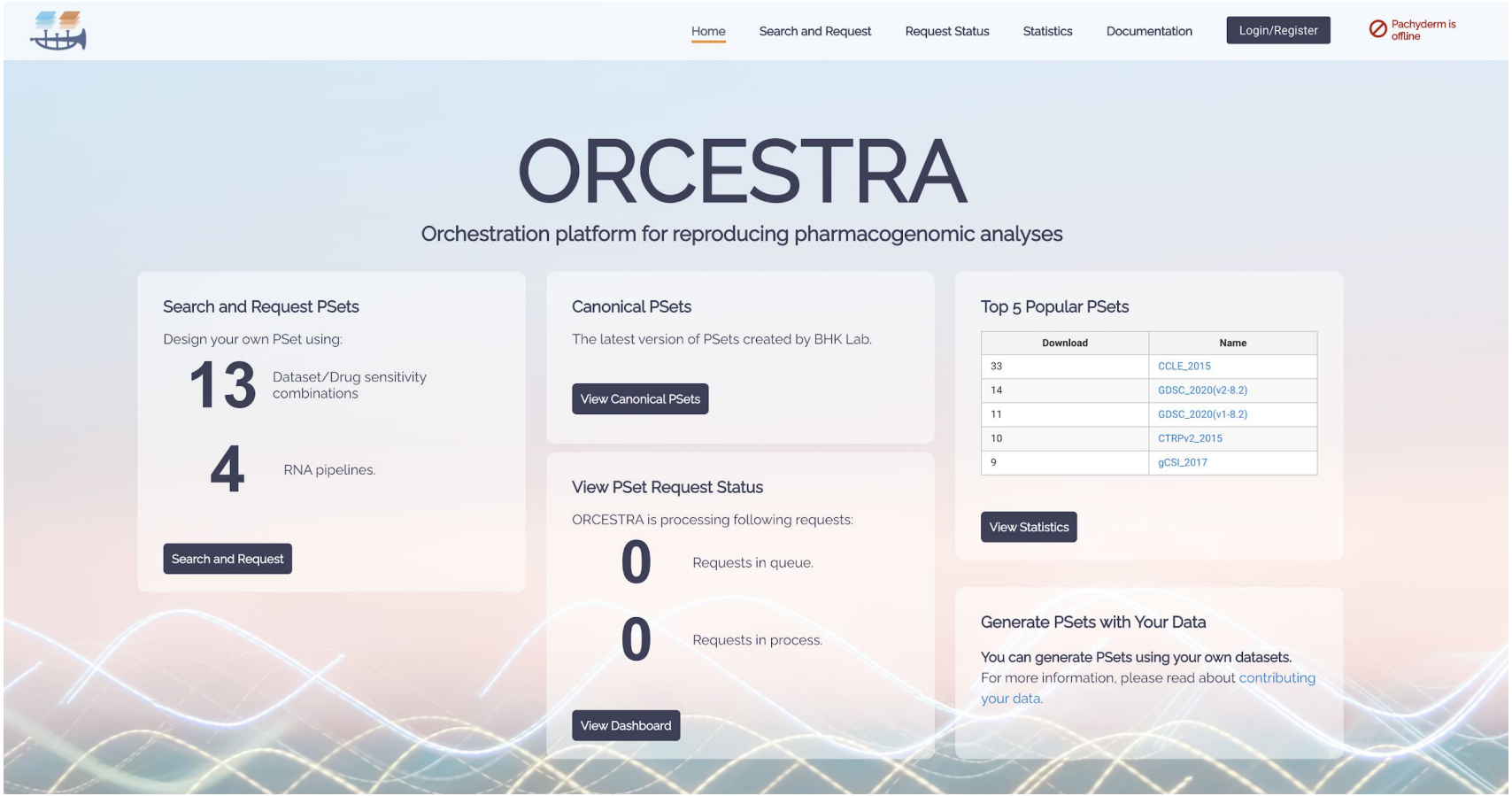
BHK LAB

ORCESTRA

Research reproducibility is also relevant for data processing

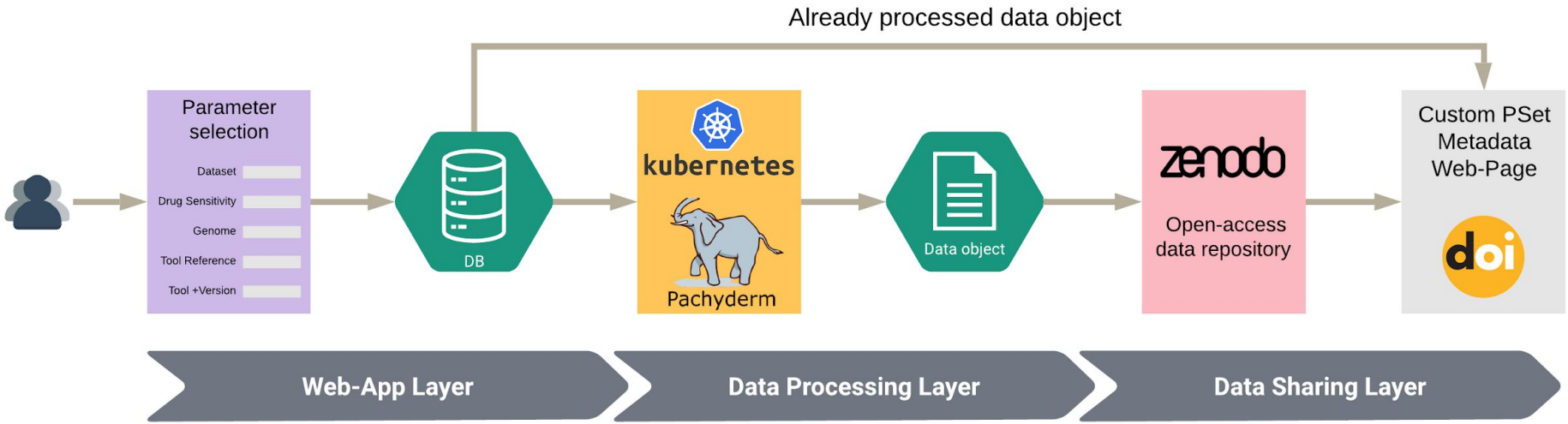
BHK LAB

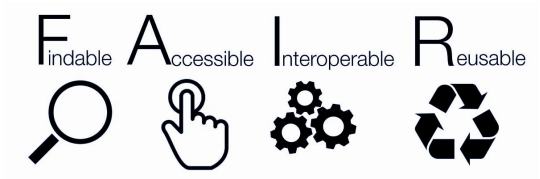
Research reproducibility is also relevant for data processing



BHK LAB

Reproducible data processing: ORCESTRA workflow





➔ Maximize cross-dataset operability by standardizing processing pipelines and compound annotations

BHK LAB

ORCESTRA for pharmacogenomics