Fred Hutch Cancer Center

Translational Data Deep Dive

Building Computational Workflows

March 11th, 2026, DaSL Lounge and Teams

We are recording…

What is a Deep Dive?

A hands-on, interactive quarterly series to help build practical data science skills at Fred Hutch

Learn Concepts

*Understand the **what**, **why**, and **when** behind data science approaches.*

See the Tools in Action

***Live demonstrations** using Hutch-supported tools and infrastructure.*

Ask & Engage

These sessions are interactive ask questions in person or online using Slido.

Get your hands dirty!

Try things yourself during the co-working time

Who we are?

And why should we be teaching you about computational workflows?

Sita: A geneticist by training, a computational biologist by choice, I have used WDL's for bulk genomics and single-cell analysis.

Taylor Firman: A bioinformatics software engineer that has used Bash/SLURM/Python in biophysics & genomics, but is much happier having discovered WDL and Docker.

Emma Bishop: A bioinformatician who learned workflows to run single-cell pipelines in the cloud and now uses WDL instead of bash scripts to analyze T cells.

Who else is here to help today

Scientific Computing

- Dan Tenenbaum
- Michael Gutteridge

PROOF Workbench *(A website to run WDL workflows on the Hutch cluster)*

- Scott Chamberlain
- Sean Kross

Training Team

- Chris Lo
- Ted Laderas

Cirro

- Sam Minot

Before we begin…

Some housekeeping to make this session work for everyone

- You can find the materials associated with this Deep Dive
  - Deep Dive Page
  - GitHub

- We will be using Slido
  - Just go to slido.com and enter this code **#8758973**
  - More details on how to use Slido here

- We follow the Training Teams DaSL Participation guidelines

Learning Objectives

Recognize When Workflows Are Useful

*Learn **when** a computational workflow is the right solution for your analysis*

Understand the Structure of a WDL

*Learn how to read and interpret the key components of **WDL**, a language to write workflows in*

Don't Build Workflows from Scratch

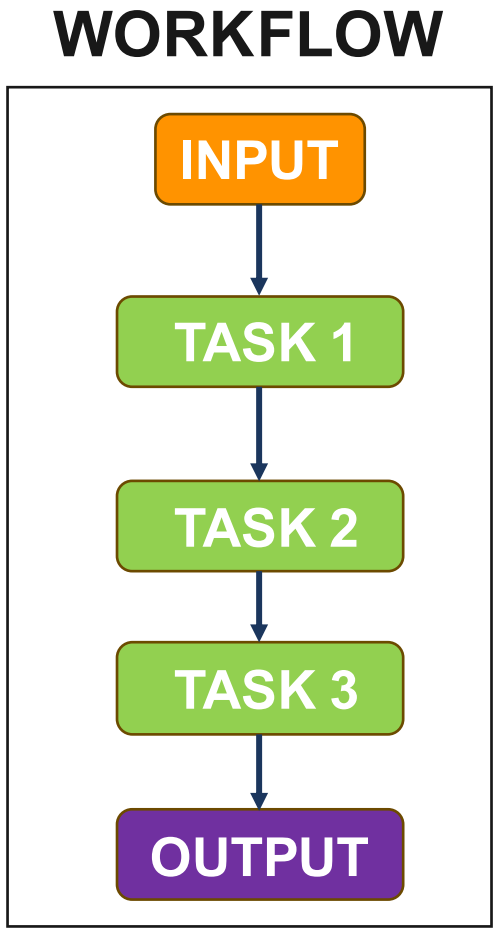
*Know about how you can leverage resources like the **WILDS WDL Library** to assemble your own workflow*

Learn how to run workflows at Fred Hutch

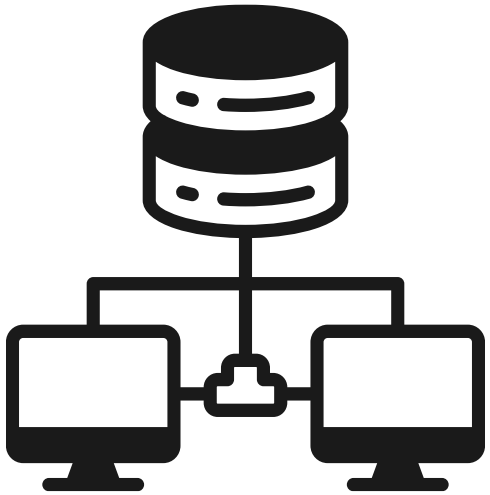
*See how workflows can be run using **PROOF Workbench** and Hutch infrastructure*

What is a Workflow?

A workflow describes all the steps needed to turn input data into results.

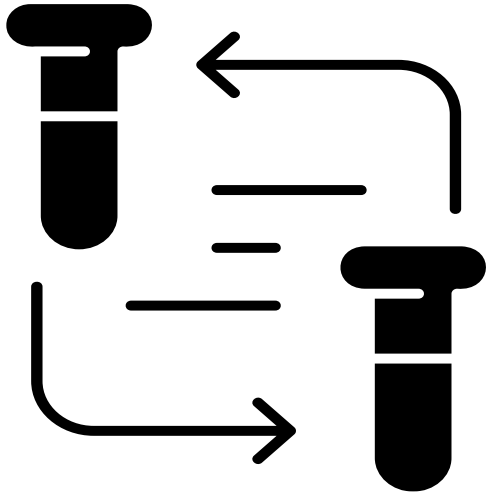


What are the advantages of writing and running a workflow?



Scalable

Many samples

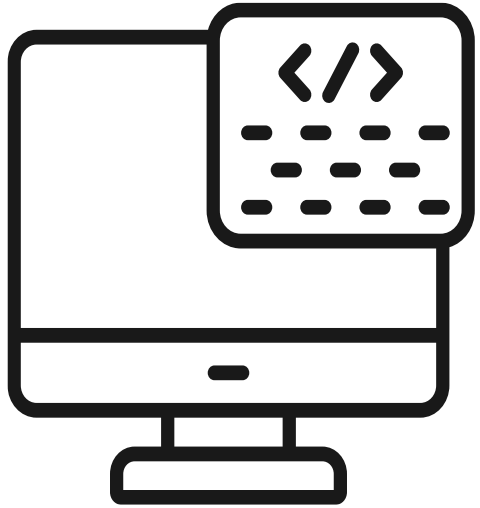


Reproducible

Defined steps

↓

Same results



Shareable

Reusable across teams

When to use a workflow?

Make a Workflow When…	DON'T Make One When…
It's **repetitive**	You're **still figuring things out**
It's **multi-step**	It's **exploratory**
It's **multi-sample**	It's **tiny**
It needs to be **reproducible**	It's **temporary**

So how do you write a workflow?

There are special workflow languages (code) in which you can give instructions… we like

Readable

workflows look like step-by-step instructions

Modular

tools are separated into reusable tasks

Built for genomics pipelines

designed for large bioinformatics analyses

How do I get started with WDL workflows?

You will usually need 3 files

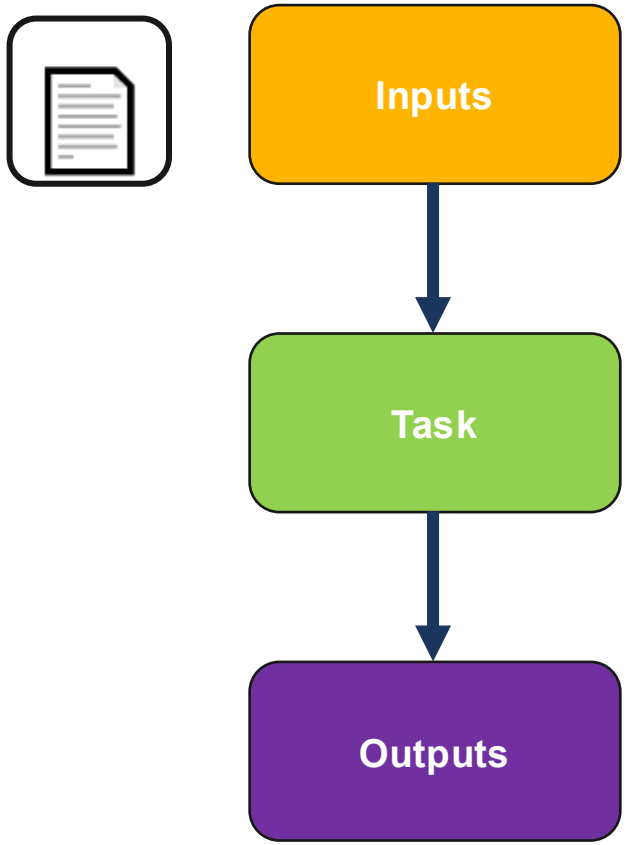
 **Ingredients** *What data to use*
`inputs.json`
Specifies **inputs**
File paths (FASTQs, BAMs, etc.)
Defines any input values for a task

 **Equipment** *What resources to use*
`options.json`
How many times a task should retry if it fails
Should you reuse previous results
Where outputs are saved

 **The Recipe** *What tasks to run*
`workflow.wdl`
Defines the tasks
Connects inputs → tasks → outputs
Declares dependencies between steps

hello_world **Workflow**

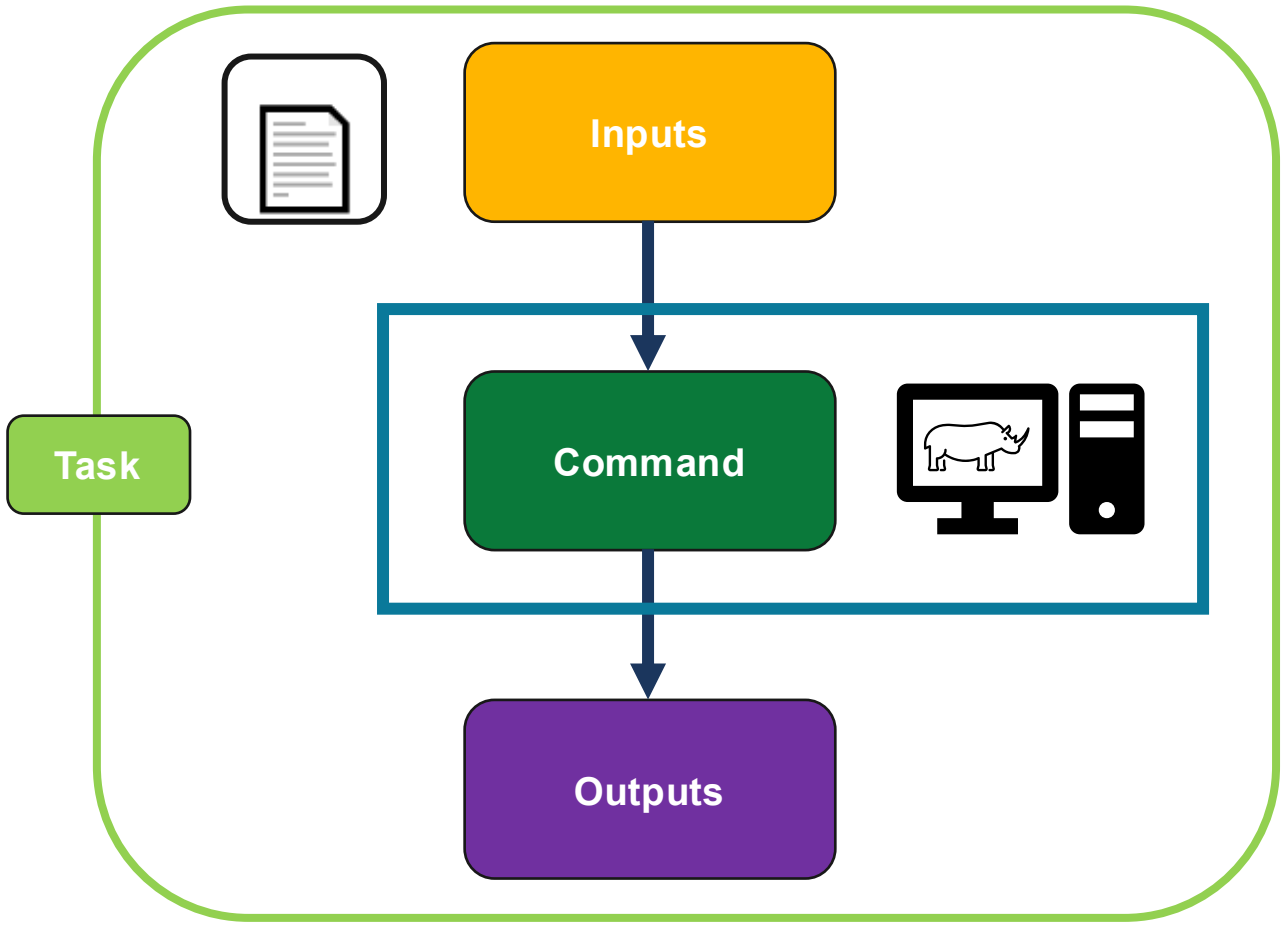
Human-readable overview of what's happening




```
workflow hello_world {
  meta {
    description: "Writes a personalized greeting"
    outputs: {
      out_file: "A file containing a greeting"
    }
  }

  parameter_meta {
    your_name: "Name to greet"
    your_number: "Your favorite number"
  }

  input {
    String your_name = "Fred"
    Int your_number = 29
  }

  call greeting {
    input:
      name = your_name,
      number = your_number
  }

  output {
    File out_file = greeting.out
  }
}
```

“Hello World” Task: greeting

Very similar vibes, but at the “building-block” task level:

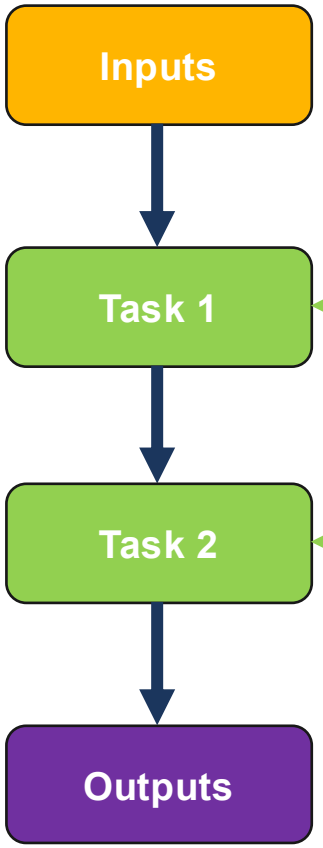



```
task greeting {
  meta {…
  }

  parameter_meta {…
  }

  input {
    String name
    Int number
  }

  command <<<
    echo "Hello, ~{name}!" > "~{name}.txt"
    echo "Your favorite number is ~{number}" \
      >> "~{name}.txt"
  >>>

  output {
    File out = "~{name}.txt"
  }

  runtime {
    docker: "getwilds/awscli:2.27.49"
    memory: "2 GB"
    cpu: 1
  }
}
```

A real-world example: ww-sra-salmon

Benefits of WDL become clear at scale…

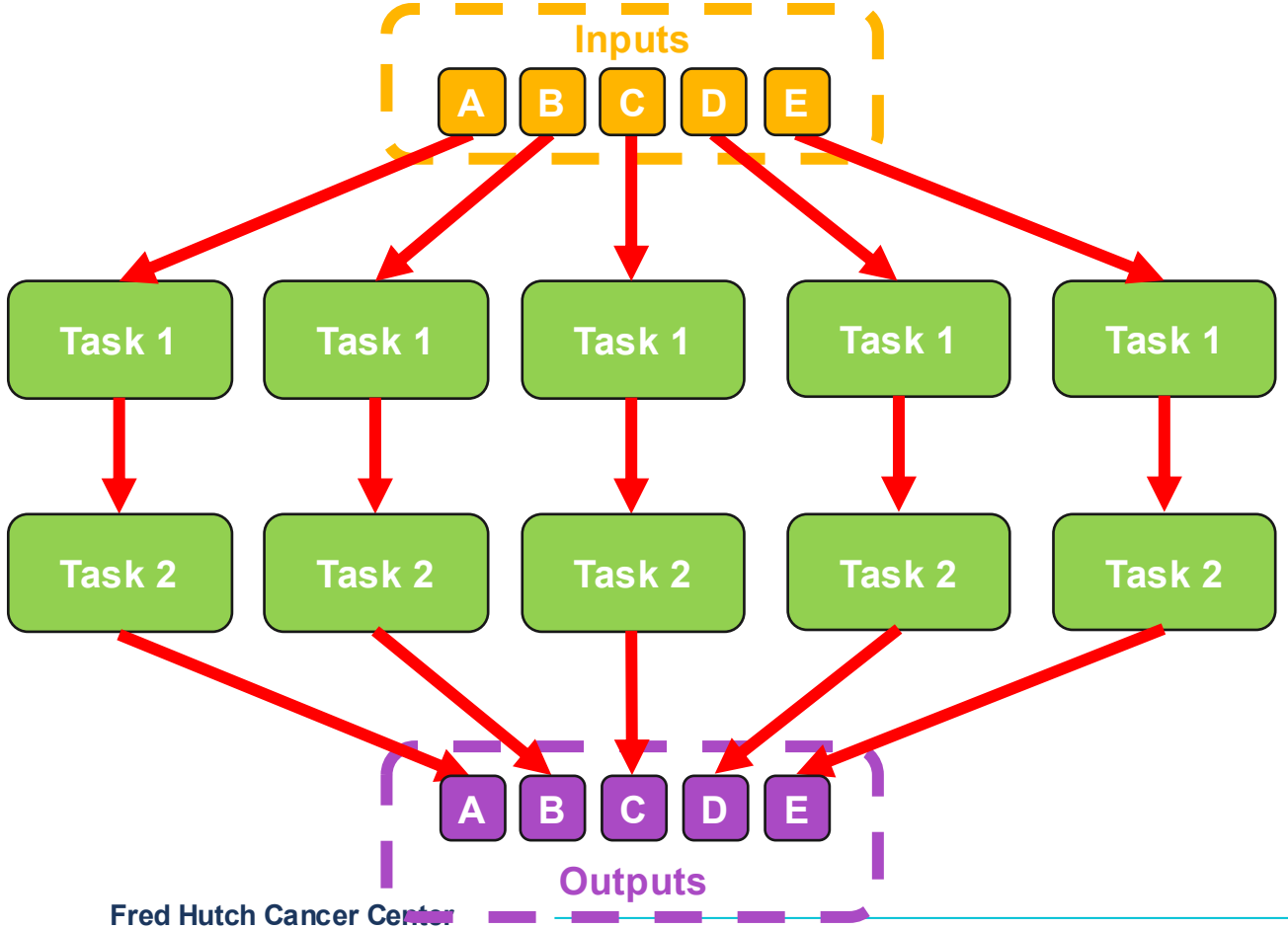
1. Specify what data I'm interested in
2. Download data from source
3. Post-process as necessary
4. Save final results locally




```
import modules/ww-sra/ww-sra.wdl as sra_tasks
import modules/ww-salmon/ww-salmon.wdl as salmon_tasks

workflow sra_salmon {
  input {
    Array[String] sra_id_list
    File salmon_index_zip
  }

  # Download FASTQ files from SRA
  call sra_tasks.fastqdump { input: sra_id = id }

  # Quantify each sample with Salmon
  call salmon_tasks.quantify { input:
      salmon_index_dir = salmon_index_zip,
      fastq_r1 = fastqdump.r1_end,
      fastq_r2 = fastqdump.r2_end,
  }

  output {
    Array[File] quant_dirs = quantify.salmon_quant_dir
  }
}
```

A real-world example: ww-sra-salmon

Benefits of WDL become clear at scale…




```
import modules/ww-sra/ww-sra.wdl as sra_tasks
import modules/ww-salmon/ww-salmon.wdl as salmon_tasks

workflow sra_salmon {
  input {
    Array[String] sra_id_list
    File salmon_index_zip
  }

  # Scatter across nodes for max parallelization
  scatter ( id in sra_id_list ){

    # Download FASTQ files from SRA
    call sra_tasks.fastqdump { input: sra_id = id }

    # Quantify each sample with Salmon
    call salmon_tasks.quantify { input:
        salmon_index_dir = salmon_index_zip,
        fastq_r1 = fastqdump.r1_end,
        fastq_r2 = fastqdump.r2_end,
    }
  }

  output {
    Array[File] quant_dirs = quantify.salmon_quant_dir
  }
}
```

Customize the workflows as needed

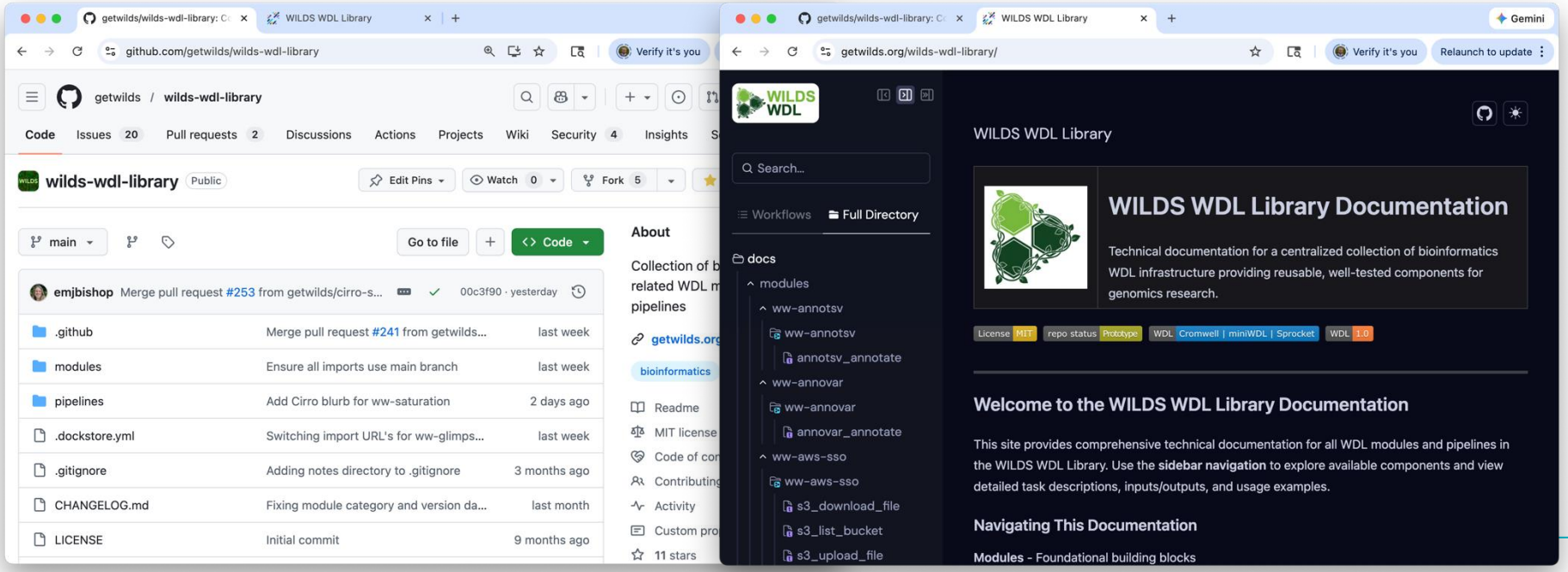
Straightforward to customize a workflow to suit your research needs

I want to…	Solution	Bioinformatics Example
…provide my own datasets	Update the inputs!	Array[File] → [file1.txt, …]
…use a different tool	Swap out the module import!	ww-salmon → ww-star
…add an extra tool	Import another module!	RNAseq → ww-deseq2
…save my results to S3	There's a module for that!!	ww-aws-sso (Upload & Download)
…make a new module	We'll help you build one!!!	WILDS WDL Library WILDS Docker Library

Easiest way to write a WDL: use someone else's!

Plenty of existing WDL workflows out there… including right here at Fred Hutch!

- WILDS WDL Library: curated collection of pre-built, well-tested, well-documented workflows
  - We can help you build one via the WILDS WDL Development Program!!



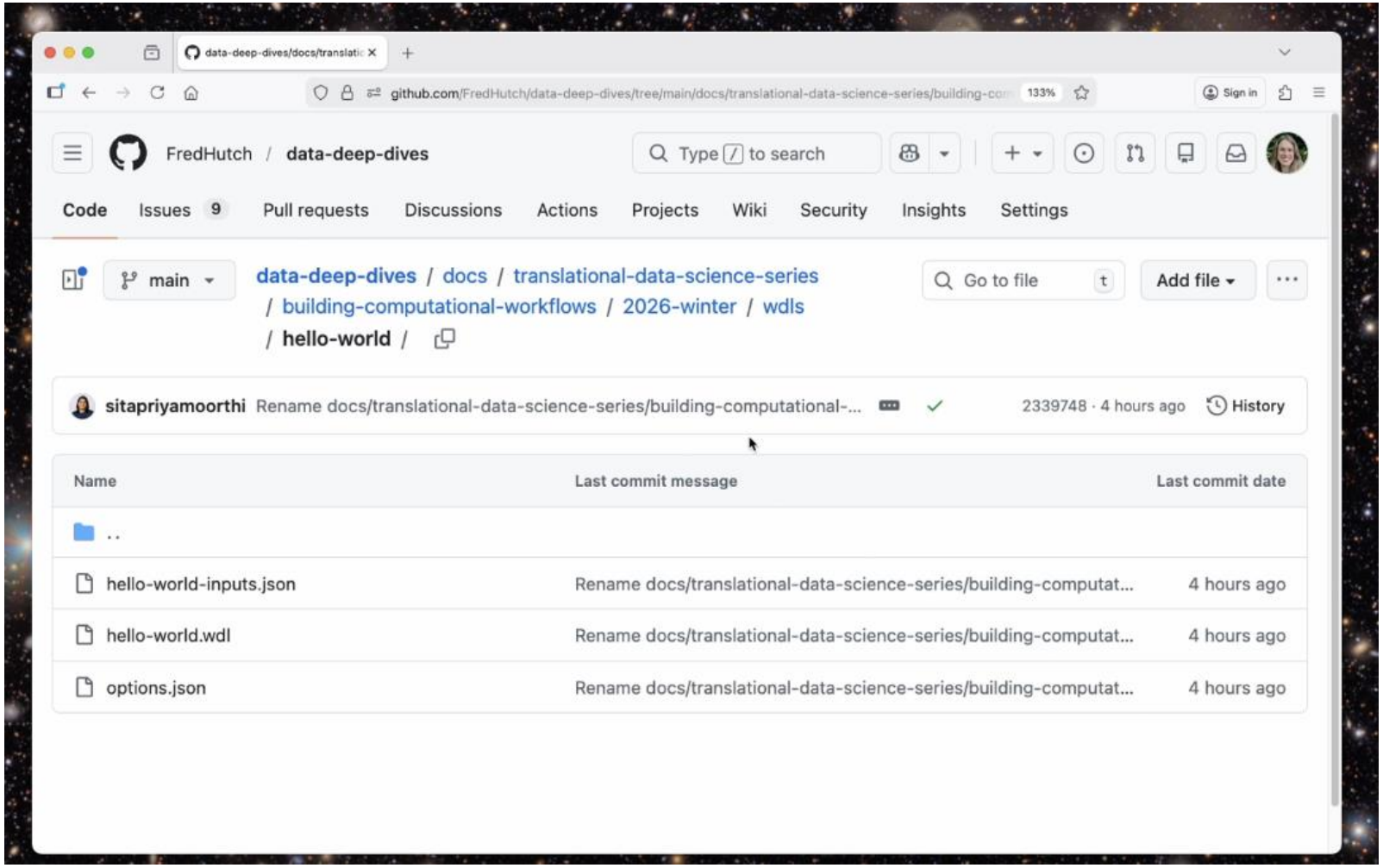
More details here!!!

Other Resources

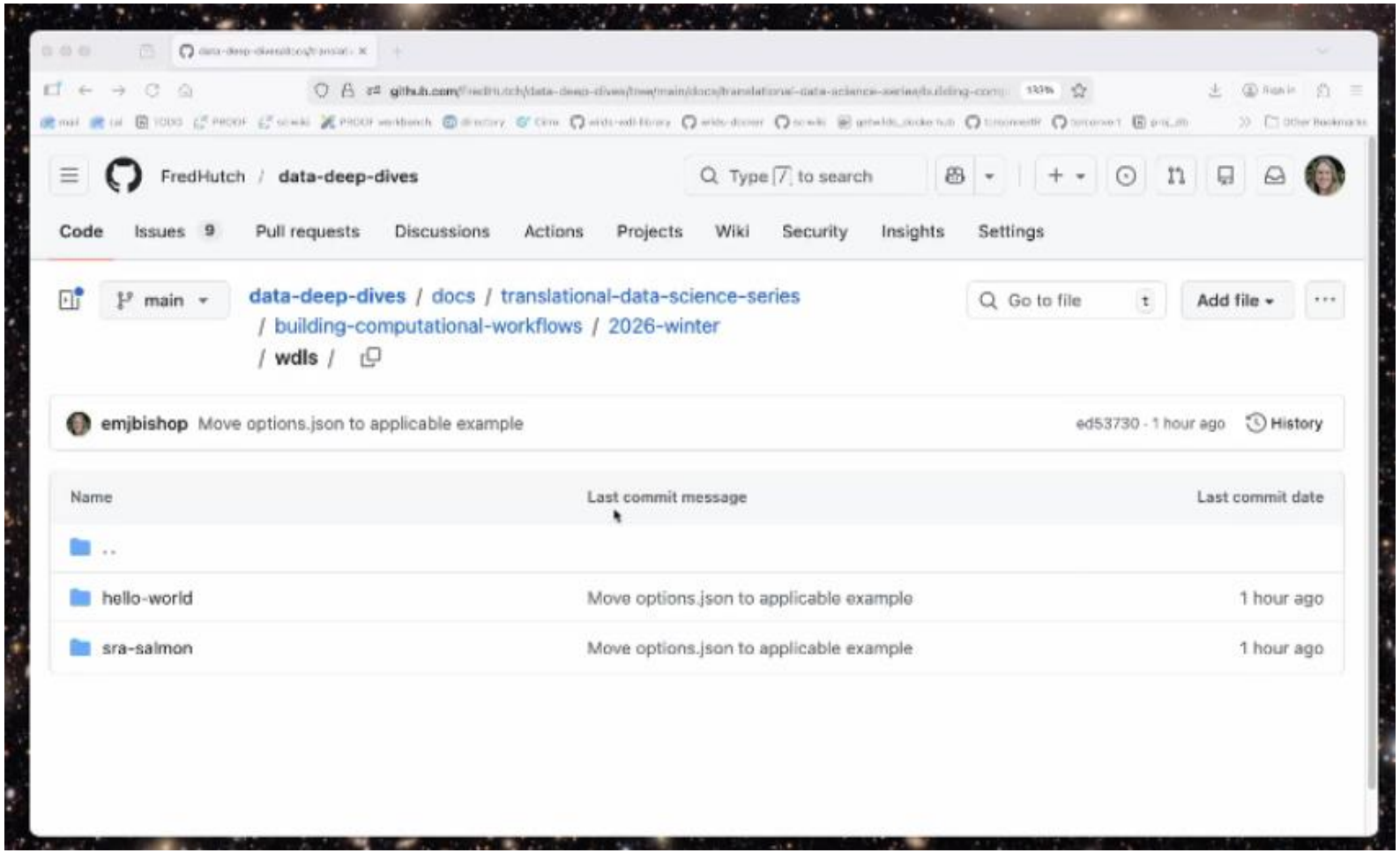
Broad's WARP
GATK Workflows
BioWDL
Dockstore

Let’s see workflows in action!

Demo: hello-world.wdl



Demo: sra-salmon.wdl

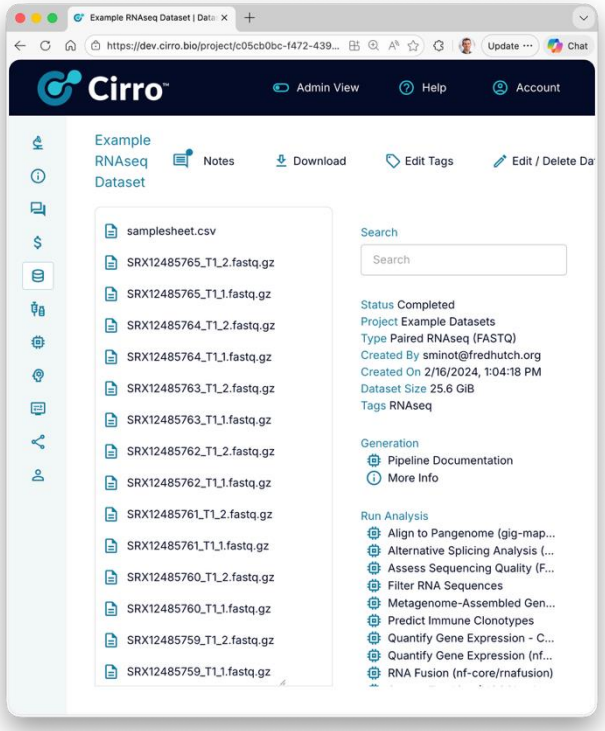


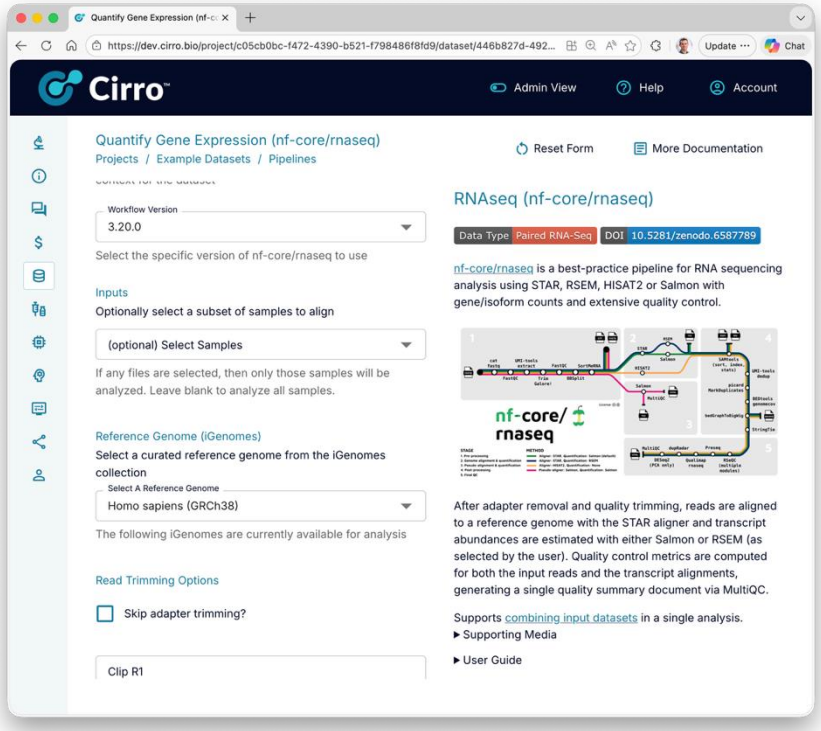
Running WDLs at Fred Hutch

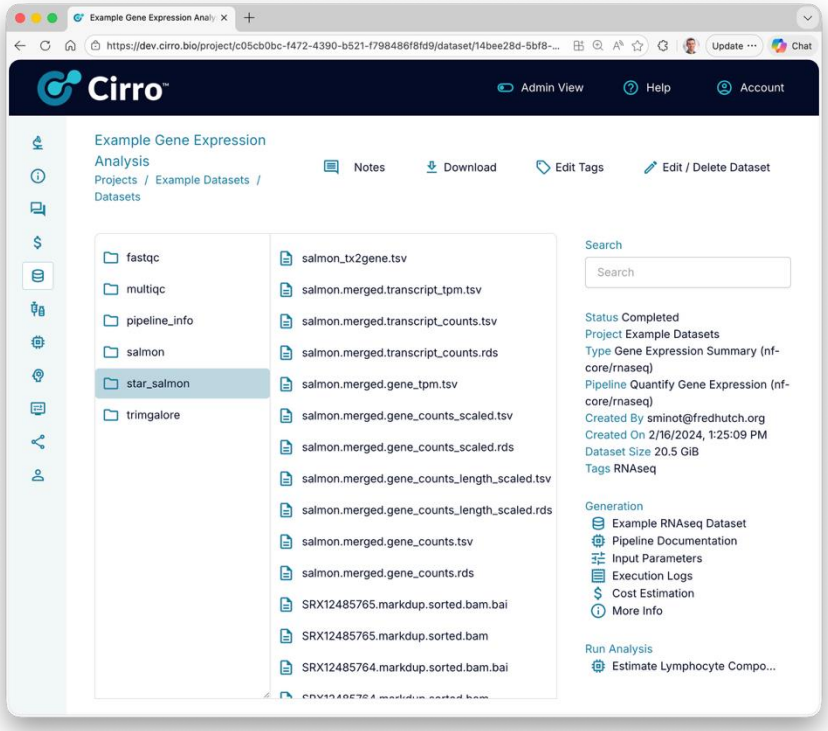
Tool	Interface	Can run	Good for
Sprocket	Command line	Any WDL	Development, running workflows manually
PROOF Workbench	Website	Any WDL	Running at scale, validating before moving to the cloud
Cirro	Website	Cirro WDLs	Using established pipelines

Also popular: miniWDL, Terra, AnVIL

Deploying Workflows in the Cirro Data Platform







1. Ingest Raw Data

2. User Friendly Pipeline Execution

3. Pipeline Outputs + Provenance

Tips for Success

- **Test then scale:** Start with a few samples first
- **Spot-check outputs:** Ensure they look reasonable
- **Check workflow status:** Confirm it started and didn’t fail
- **Read logs to troubleshoot:** Scan for error messages to get context

💬 Reach out on Slack, we've probably seen your issue before

Fred Hutch WDL Resources

WILDS WDL Library
Validated WDLs

WILDS Docker Library
Validated Docker images

PROOF Workbench
Run WDLs on the cluster through a website

Fred Hutch Data Science Help

FH-Data Slack
Crowdsourcing, Q&A

SciComp
Get help with the cluster, the cloud, data storage

SciWiki
Useful knowledgebase

Data House Calls
Short meetings for personalized help

Data Science Lab Training Spring Quarter, 2026

Courses

- Bash for Bioinformatics*
- Machine Learning for Python*(new!)
- Bioconductor for Genomics*
- Intro to R

**has prerequisites*

Workshops

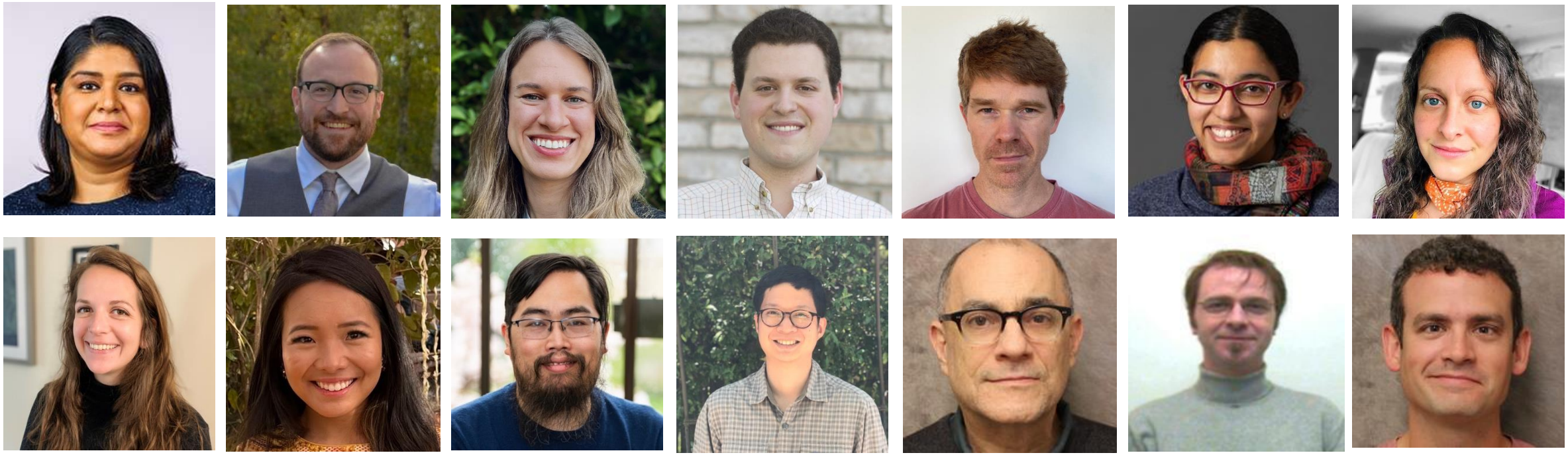
- Better Tables
- Intro to Command Line
- Intro to Cluster Computing

Registration opens soon!
Quarter begins April 28

For more info*: send a blank email to* dasltraining-join@lists.fhcrc.org

Thank you to everyone who made this Deep Dive possible!

And thank you all for participating!!



Fred Hutch Office of the Chief Data Officer (OCDO)

Fred Hutch Scientific Computing (SciComp)

Fred Hutch Cancer Center

Thank you