



# Bringing MultiQC into a new era

Recent updates and what to expect  
from the roadmap towards v2.0



# Modern software engineering - for science

## Genomic analysis, simplified

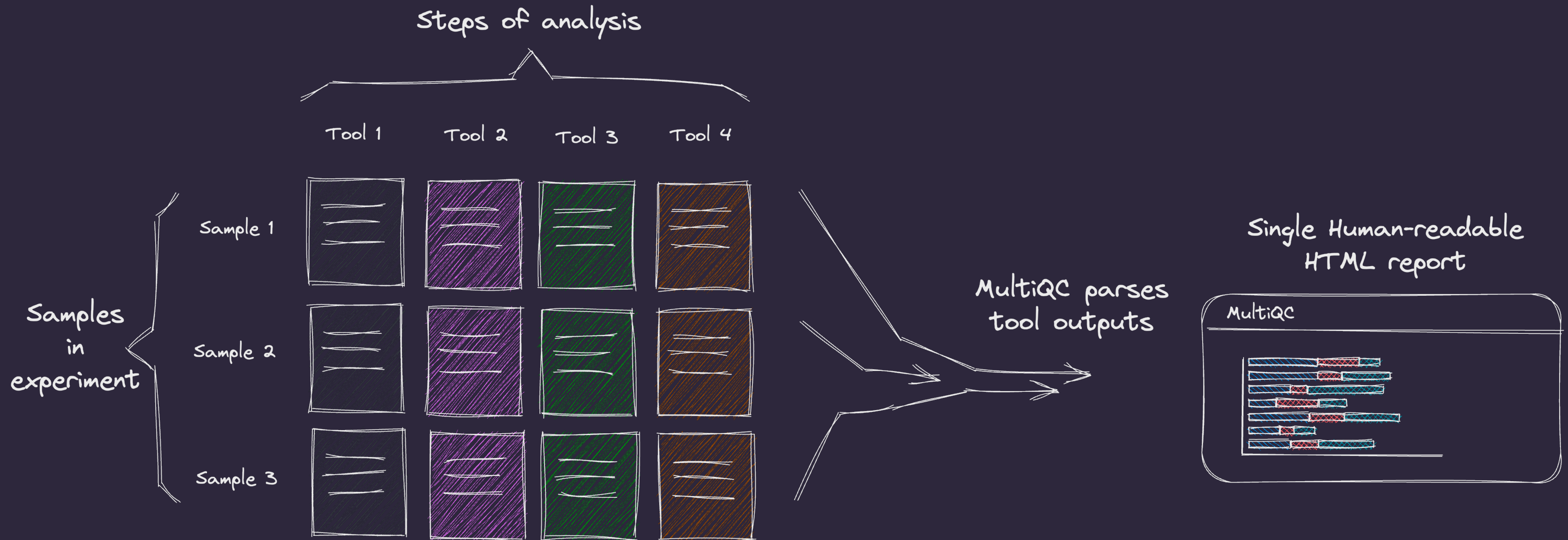
Easily aggregate results across bioinformatics studies of any size where ensuring accurate quality control across datasets is critical.

MultiQC integrates with the Seqera Platform enabling analysts to easily access reports.

MultiQC is used in thousands of high-quality pipelines and is trusted by leading private and public research institutions worldwide.



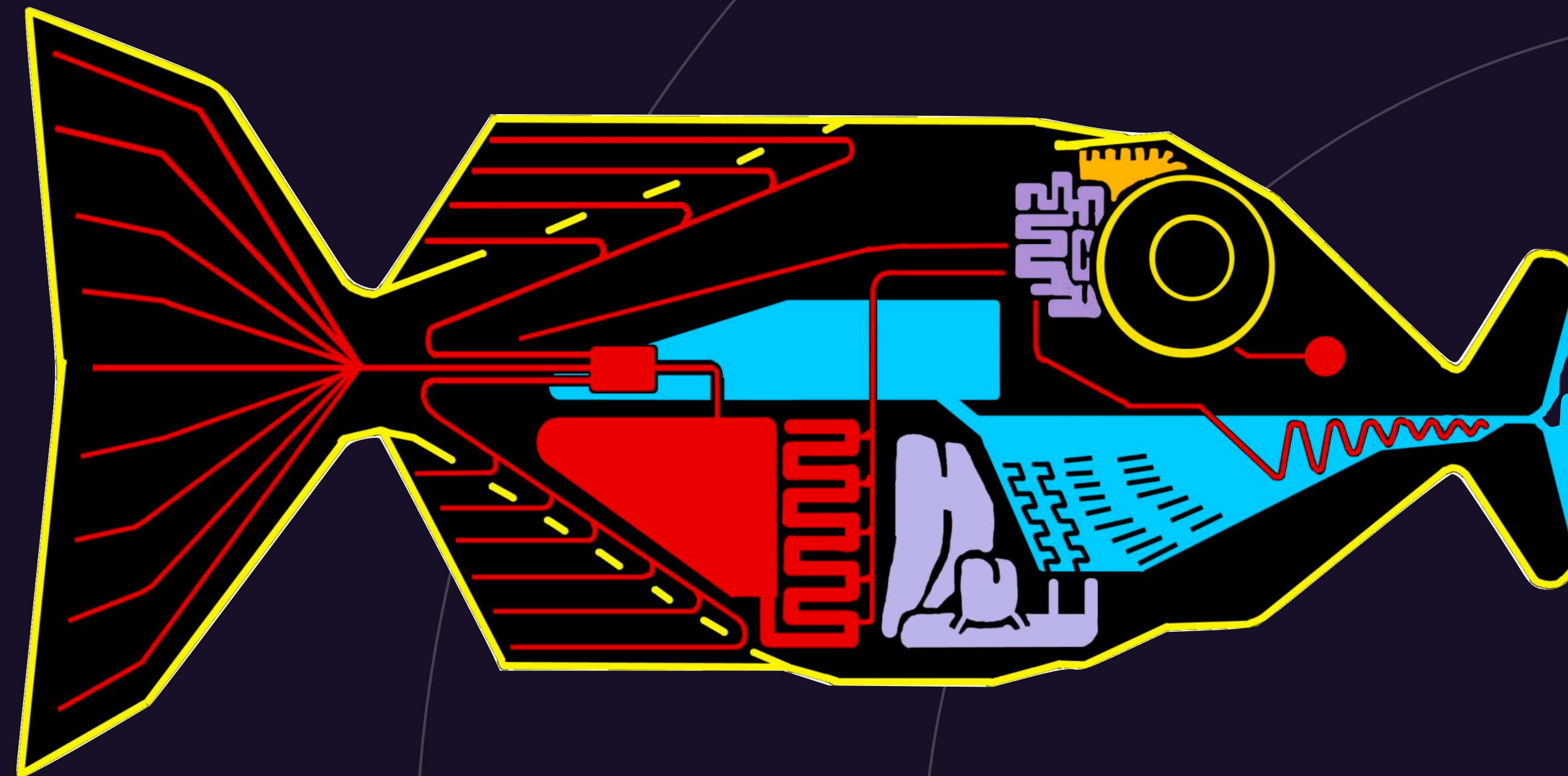
# Modern software engineering - for science





# Modern software engineering - for science

Bioinformatics  
log outputs



Human-  
readable  
report

MULTIQC



# MultiQC by the numbers

**135**

Supported tools

**1,000,000**

Downloads from PyPI

**1,057**

GitHub Stars ★

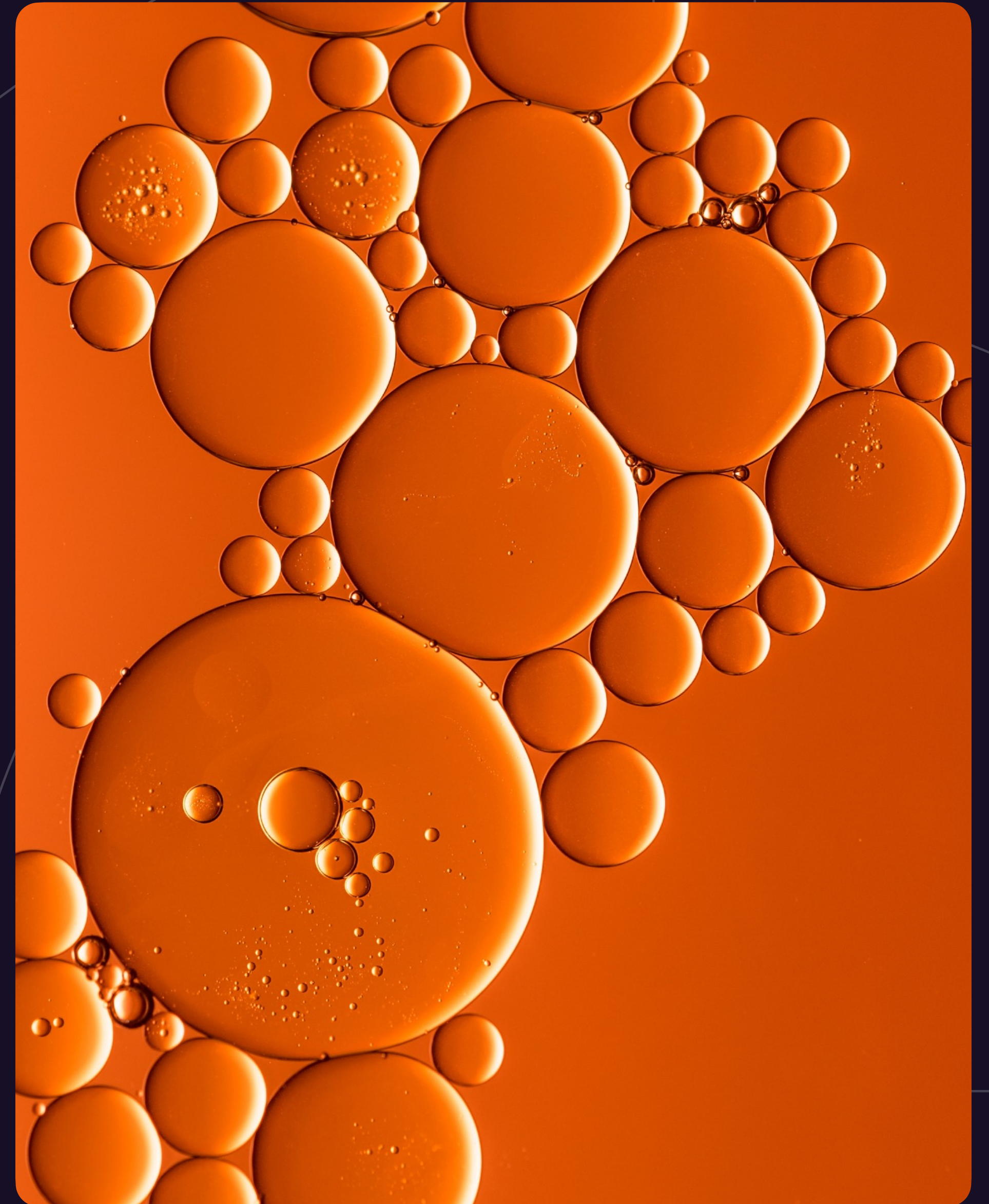
**~1/sec**

MultiQC runs



# Recent updates

What's new in the world of  
MultiQC





# Vladislav Savelyev

## Vlad joins the MultiQC team at Seqera

- Key contributor to MultiQC over the years
- Moved from Melbourne, Australia
- Picking up MultiQC development after several years of minimal maintenance





MultiQC file search

**720% faster**



# Citations and DOIs

# 4000

Journal citations

The **NASA Twins Study**: A multidimensional analysis of a year-long human spaceflight

Attenuation of clinical and immunological outcomes during **SARS-CoV-2** infection by ivermectin

**In vivo base editing** rescues Hutchinson–Gilford progeria syndrome in mice

Biofilm formation of *Pseudomonas aeruginosa* **in spaceflight** is minimized on lubricant impregnated surfaces

# Citations and DOIs

```
star.py

def __init__(self):
    # Initialise the parent object
    super(MultiqcModule, self).__init__(
        name="STAR",
        anchor="star",
        href="https://github.com/alexdobin/STAR",
        info="is an ultrafast universal RNA-seq aligner.",
        doi="10.1093/bioinformatics/bts635",
    )
```

## STAR

**STAR** is an ultrafast universal RNA-seq aligner. DOI: [10.1093/bioinformatics/bts635](https://doi.org/10.1093/bioinformatics/bts635).



# Citations and DOIs

```
def __init__(self):  
    # Initialise the parent object  
    super(MultiqcModule, self).__init__(  
        name="STAR",  
        anchor="star",  
        href="https://github.com/alexdobin/STAR",  
        info="is an ultrafast universal RNA-seq aligner.",  
        doi="10.1093/bioinformatics/bts635",  
    )
```

## STAR

STAR is an ultrafast universal RNA-seq aligner. DOI: [10.1093/bioinformatics/bts635](https://doi.org/10.1093/bioinformatics/bts635).

# Automatic version parsing

# Automatic version parsing

```
salmon.py

# Parse meta information. JSON win!
self.salmon_meta = dict()
for f in self.find_log_files("salmon/meta"):
    # Get the s_name from the parent directory
    s_name = os.path.basename(os.path.dirname(f["root"]))
    s_name = self.clean_s_name(s_name, f)
    self.salmon_meta[s_name] = json.loads(f["f"])
    self.add_software_version(self.salmon_meta[s_name]["salmon_version"], s_name)
```

**Salmon** Version: 0.9.1

Salmon is a tool for quantifying the expres

```
cmd_info.json

{
  "salmon_version": "0.9.1",
  "index": "git_repositories/ref-txome/athaliana/ref
  "libType": "A",
  "mates1": "git_repositories/ref-txome/athaliana/da
  "mates2": "git_repositories/ref-txome/athaliana/da
```



# Automatic version parsing

```
salmon.py

# Parse meta information. JSON win!
self.salmon_meta = dict()
for f in self.find_log_files("salmon/meta"):
    # Get the s_name from the parent directory
    s_name = os.path.basename(os.path.dirname(f["root"]))
    s_name = self.clean_s_name(s_name, f)
    self.salmon_meta[s_name] = json.loads(f["f"])
self.add_software_version(self.salmon_meta[s_name]["salmon_version"], s_name)
```

**Salmon** Version: 0.9.1

Salmon is a tool for quantifying the expres

```
cmd_info.json

{
  "salmon_version": "0.9.1",
  "index": "git_repositories/ref-txome/athaliana/ref
  "libType": "A",
  "mates1": "git_repositories/ref-txome/athaliana/da
  "mates2": "git_repositories/ref-txome/athaliana/da
```

# Manual version reporting

fastqc\_mqc\_versions.yaml

```
FASTQC:  
  fastqc: "0.11.9"
```

salmon\_mqc\_versions.yaml

```
SALMON_QUANT:  
  salmon: "1.10.1"
```

star\_mqc\_versions.yaml

```
STAR_ALIGN:  
  star: "2.6.1d"  
  samtools: "1.10"  
  gawk: "5.1.0"
```

## Software Versions

Software Versions lists versions of software tools extracted from file contents.

 Copy table

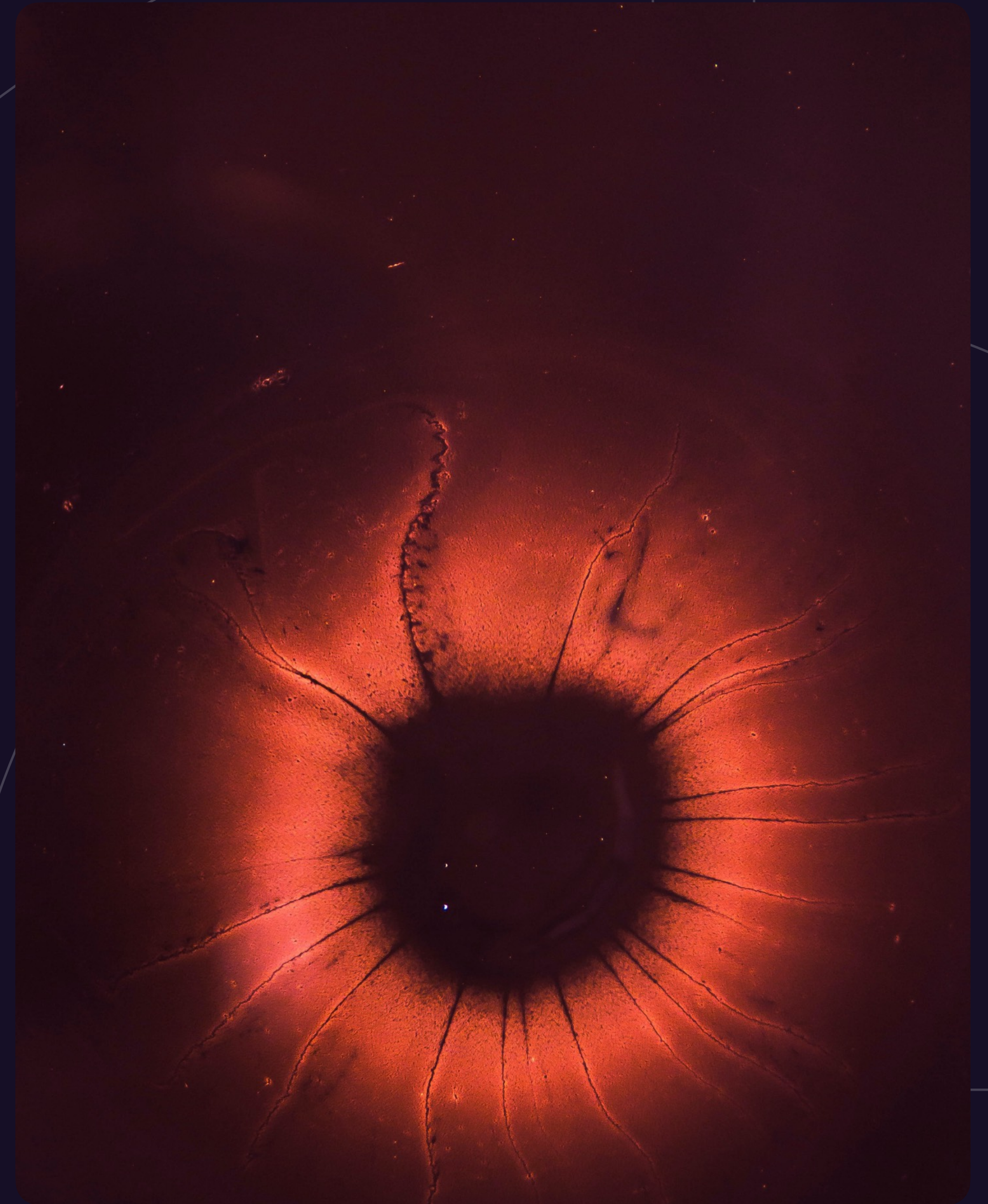
Group	Software	Version
FASTQC	fastqc	0.11.9
STAR_ALIGN	star	2.6.1d
	samtools	1.10
	gawk	5.1.0
SALMON_QUANT	salmon	1.10.1





# Roadmap

Looking ahead to v2.0

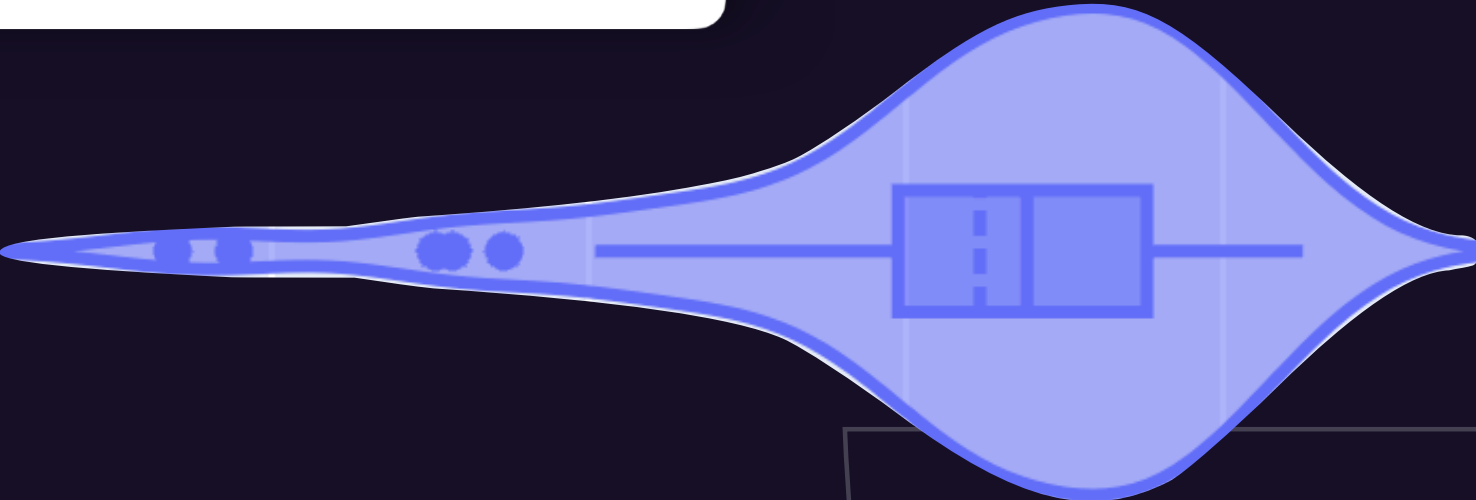
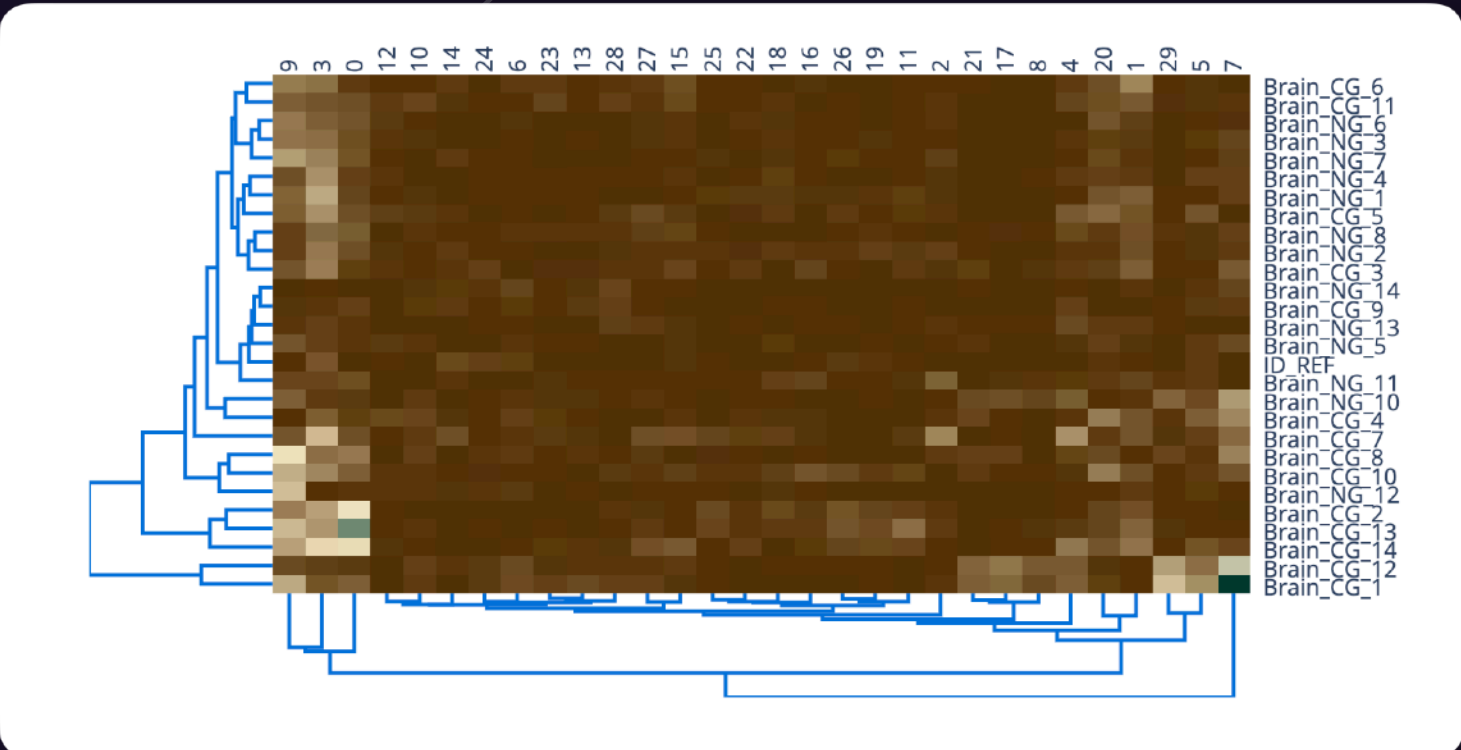
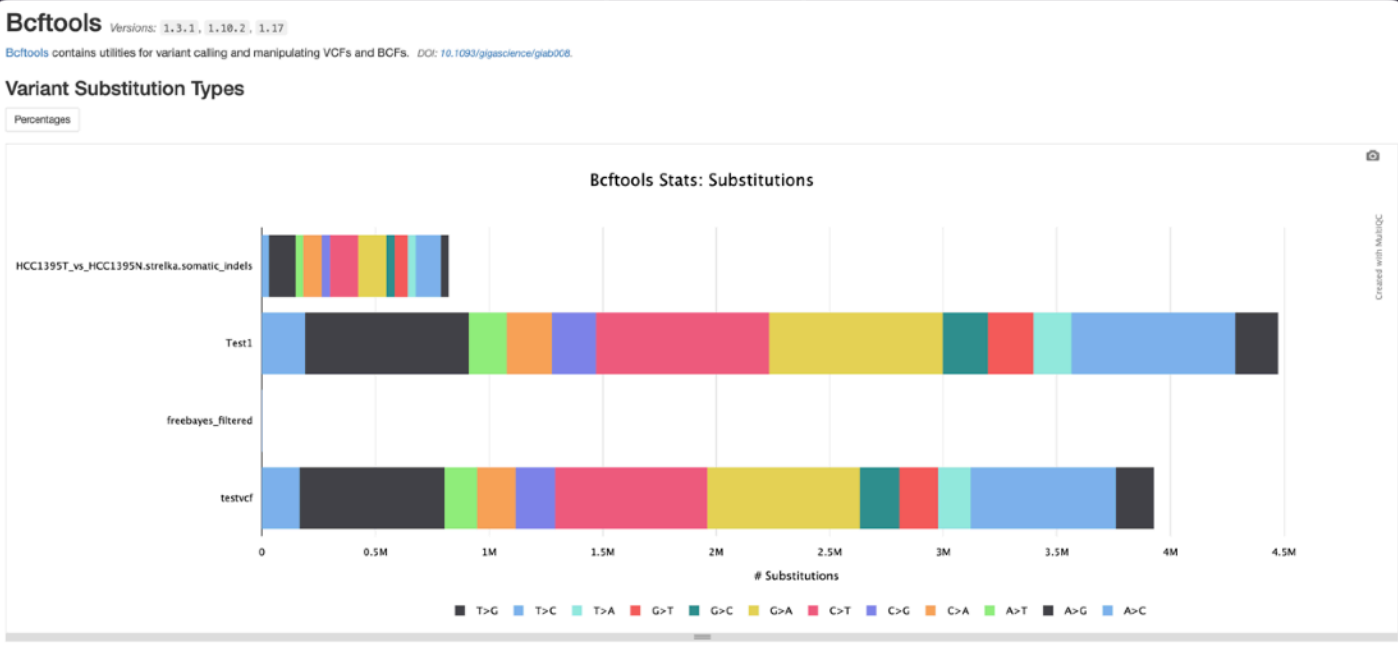




# Replacing the plotting library

## Switching HighCharts with Plotly

- Many more plot types available
- Can generate interactive plots and static images in Python

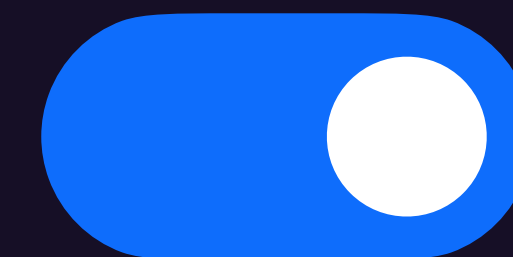




# Updates to the default template

## A fresh lick of pixels

- Rewrite the interactive tooling to work with Plotly
- Dark mode 🕶️❤️
- Updates to the JavaScript / CSS frameworks

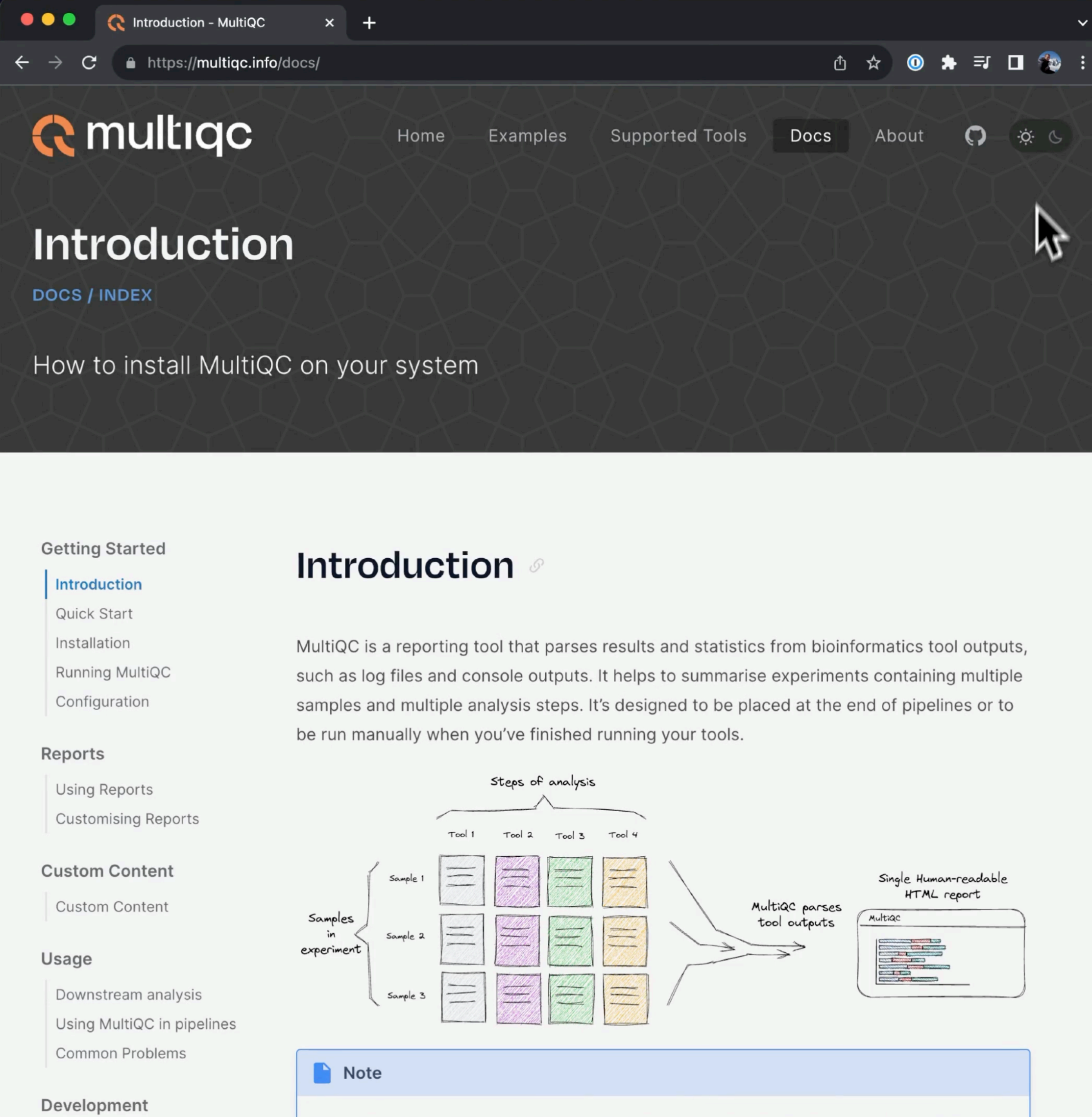


Dark mode

# Updates to the default template

## A fresh lick of pixels

- Dark mode 🕶️❤️
- Updates frameworks
- *Bonus:* Find the easter egg on [multiqc.info!](https://multiqc.info/)



The screenshot shows the MultiQC documentation website in dark mode. The browser address bar displays `https://multiqc.info/docs/`. The navigation menu includes Home, Examples, Supported Tools, Docs (active), and About. The main heading is "Introduction" with a sub-link "DOCS / INDEX". Below the heading, it says "How to install MultiQC on your system".

The content area features a sidebar on the left with the following sections:

- Getting Started
  - Introduction (active)
  - Quick Start
  - Installation
  - Running MultiQC
  - Configuration
- Reports
  - Using Reports
  - Customising Reports
- Custom Content
  - Custom Content
- Usage
  - Downstream analysis
  - Using MultiQC in pipelines
  - Common Problems
- Development

The main content area is titled "Introduction" and contains the following text:

MultiQC is a reporting tool that parses results and statistics from bioinformatics tool outputs, such as log files and console outputs. It helps to summarise experiments containing multiple samples and multiple analysis steps. It's designed to be placed at the end of pipelines or to be run manually when you've finished running your tools.

A diagram titled "Steps of analysis" illustrates the workflow. It shows a grid of 12 colored boxes representing tool outputs for 3 samples (Sample 1, Sample 2, Sample 3) across 4 tools (Tool 1, Tool 2, Tool 3, Tool 4). An arrow labeled "MultiQC parses tool outputs" points from this grid to a box labeled "Single Human-readable HTML report" which contains a sample of MultiQC output.

At the bottom of the page, there is a "Note" section.



# Core code refactoring

## Cleaner code for all

- Better use of objects and classes instead of globals
- Python 3 best practices, including the aim to have complete variable typing
- Adopt Pydantic for rich validation and support of multiple sta
- Import MultiQC into scripts and use as a library for your own purposes

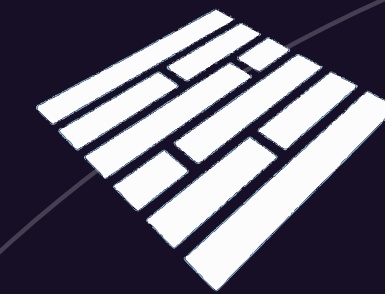


Pydantic

# Standardised file format

## MultiQC as a foundation to build upon

- Separate out the process of parsing input data and report generation
- Adopt a new standard for intermediate data, building upon Pydantic's support of formats such as Apache Parquet
- Make "Custom Content" a better supported and more mainstream feature
- Position MultiQC as a building block for larger analytics platforms



Parquet

# Generate reports in your browser

[multiqc.info/run](http://multiqc.info/run) >

multiqc

Home Examples Supported Tools Docs About

## Run MultiQC in your browser

✦ Run MultiQC in your browser on local files, with Web Assembly.

Step 1 - Choose Directory

Step 2 - Run MultiQC

Step 3 - Open Report

Staged files

- SRR3192396\_1.fastq.gz\_trimming\_report.txt
- SRR3192396\_1Log.final.out
- SRR3192396\_1Log.out
- SRR3192396\_1Log.progress.out
- SRR3192396\_1Log.std.out
- SRR3192396\_1\_fastqc.html
- SRR3192396\_1\_fastqc.zip
- SRR3192396\_1\_star\_aligned.bam\_counts.txt.summary
- SRR3192396\_1\_val\_1\_fastqc.html

```
$ multiqc .  
  
/// MultiQC | v1.16  
  
multiqc | Search path : /data  
feature_counts | Found 8 reports  
  star | Found 8 reports  
  cutadapt | Found 16 reports  
  fastqc | Found 32 reports  
multiqc | Report      : multiqc_report.html
```

- Generate reports in your browser, no installations necessary
- Point and click, doesn't use the terminal
- Uses Webassembly



# Thank you



## **Phil Ewels, PhD**

Product Manager for Open Source  
[phil.ewels@seqera.io](mailto:phil.ewels@seqera.io)

