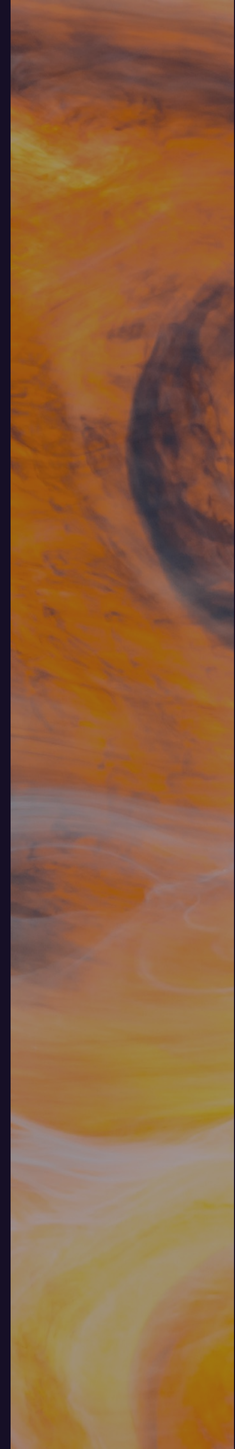
multiqc

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.

multiqc
reporting and analytics

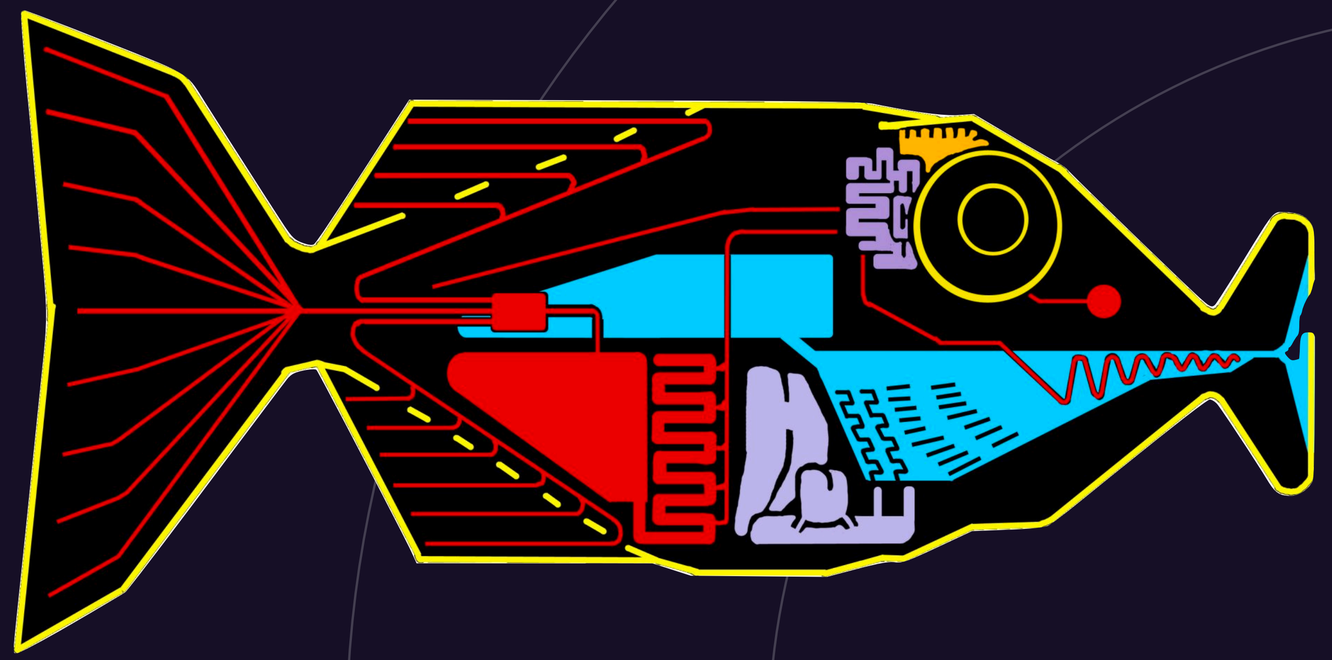
seqera

Modern software engineering – for science

seqera

Modern software engineering - for science

Bioinformatics log outputs



MULTIQC

Human-readable report

seqera

MultiQC by the numbers

135
Supported tools

1,000,000
Downloads from PyPI

1,057
GitHub Stars ★

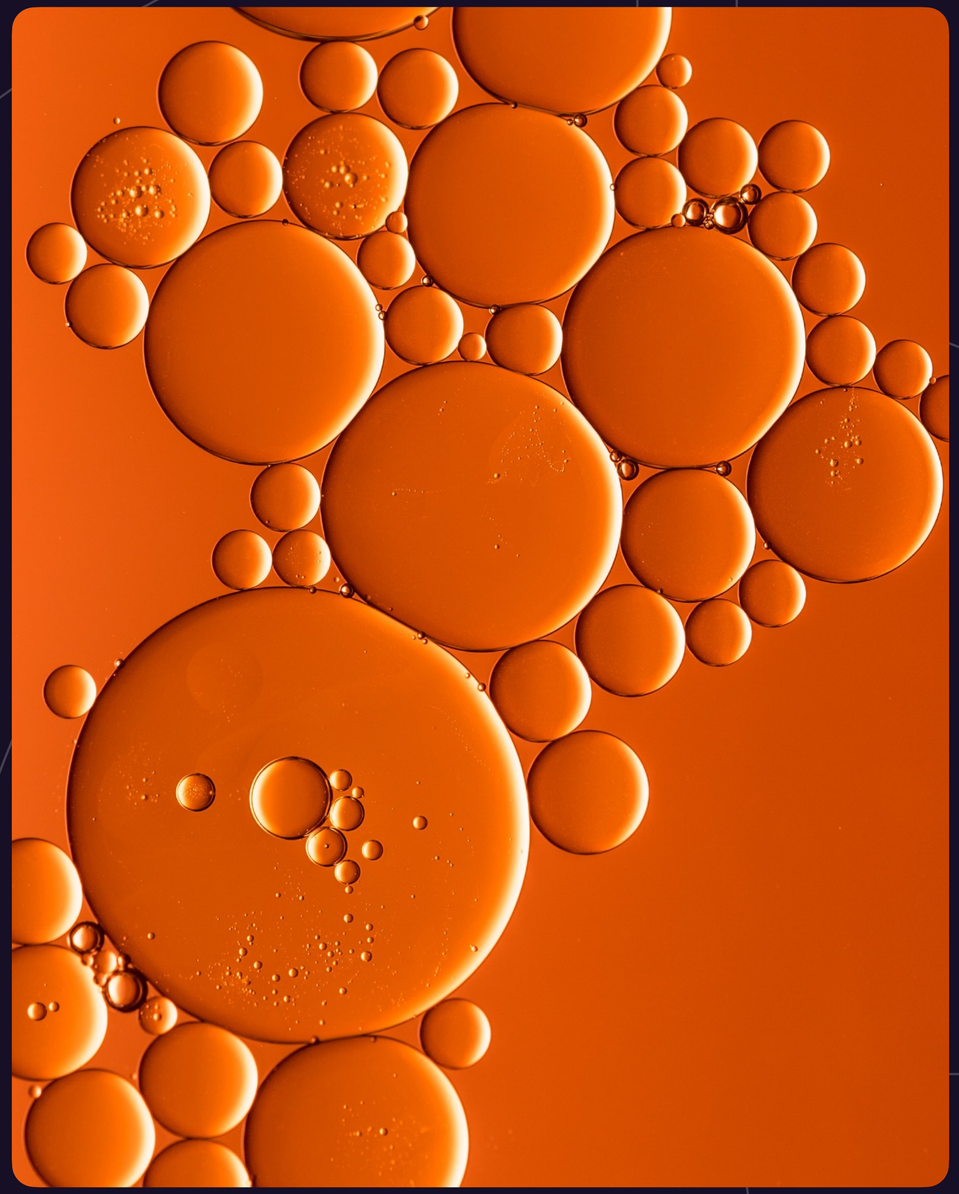
~1/sec
MultiQC runs

seqera

multiqc

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

seqera

MultiQC file search

720% faster

seqera

Citations and DOIs

4000

Journal citations

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

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

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

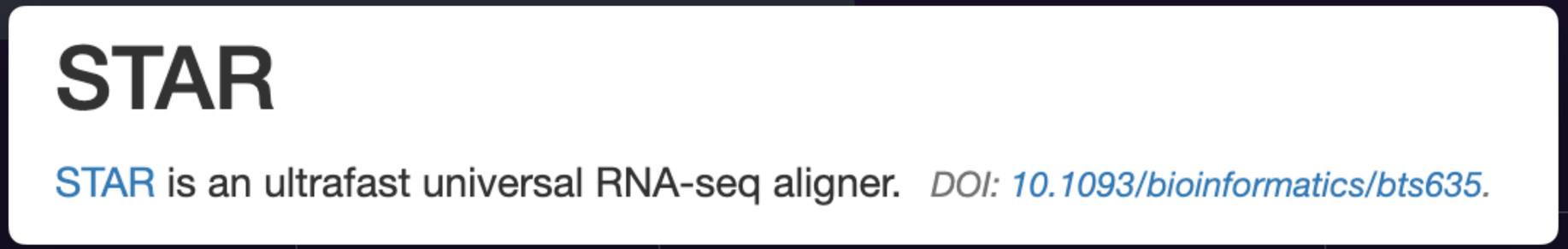
*Biofilm formation of Pseudomonas aeruginosa **in spaceflight** is minimized on lubricant impregnated surfaces*

seqera

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",
    )
```

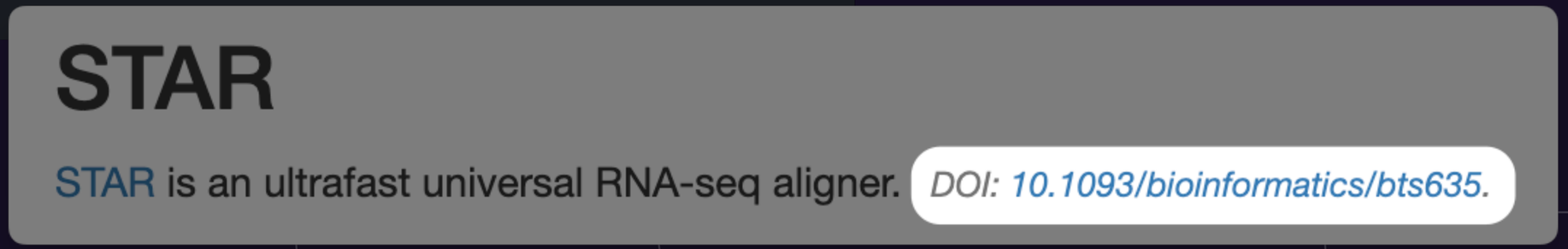


seqera

Citations and DOIs

star.py

```python
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",
    )
```



seqera

Automatic version parsing

seqera

Automatic version parsing

salmon.py

```python
# 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

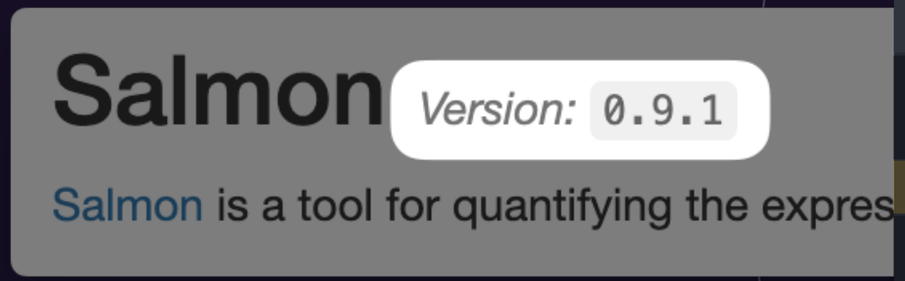
cmd_info.json

```
{
    "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

```python
# 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)
```



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

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

salmon_mqc_versions.yaml

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

star_mqc_versions.yaml

```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

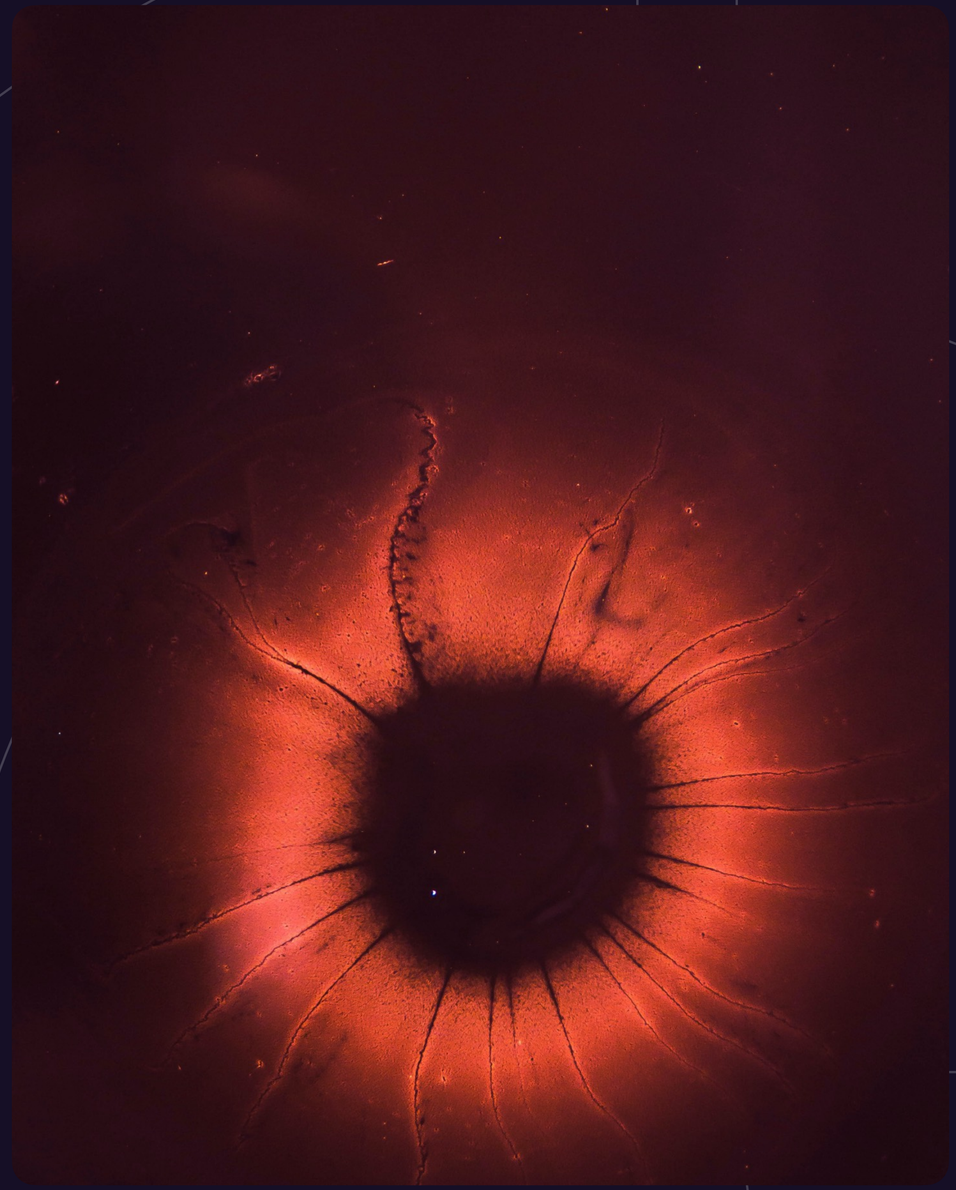
Group	Software	Version
FASTQC	fastqc	0.11.9
STAR_ALIGN	star	2.6.1d
	samtools	1.10
	gawk	5.1.0
SALMON_QUANT	salmon	1.10.1

seqera

multiqc

Roadmap

Looking ahead to v2.0

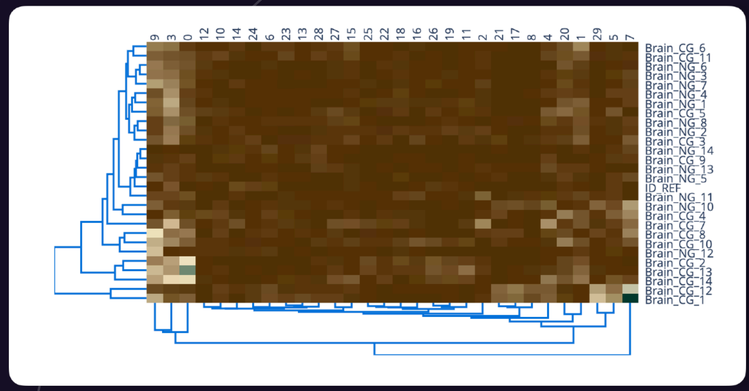


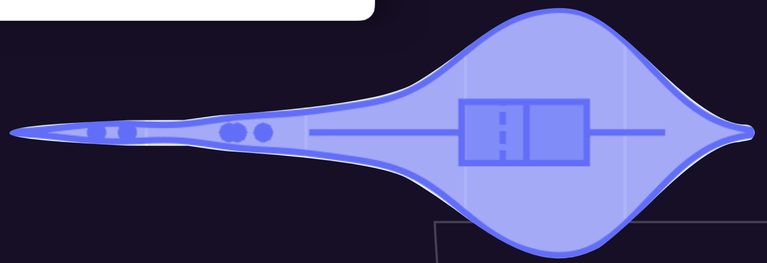
Replacing the plotting library

Switching HighCharts with Plotly

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

plotly



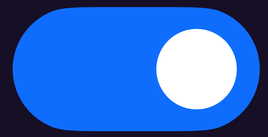


seqera

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



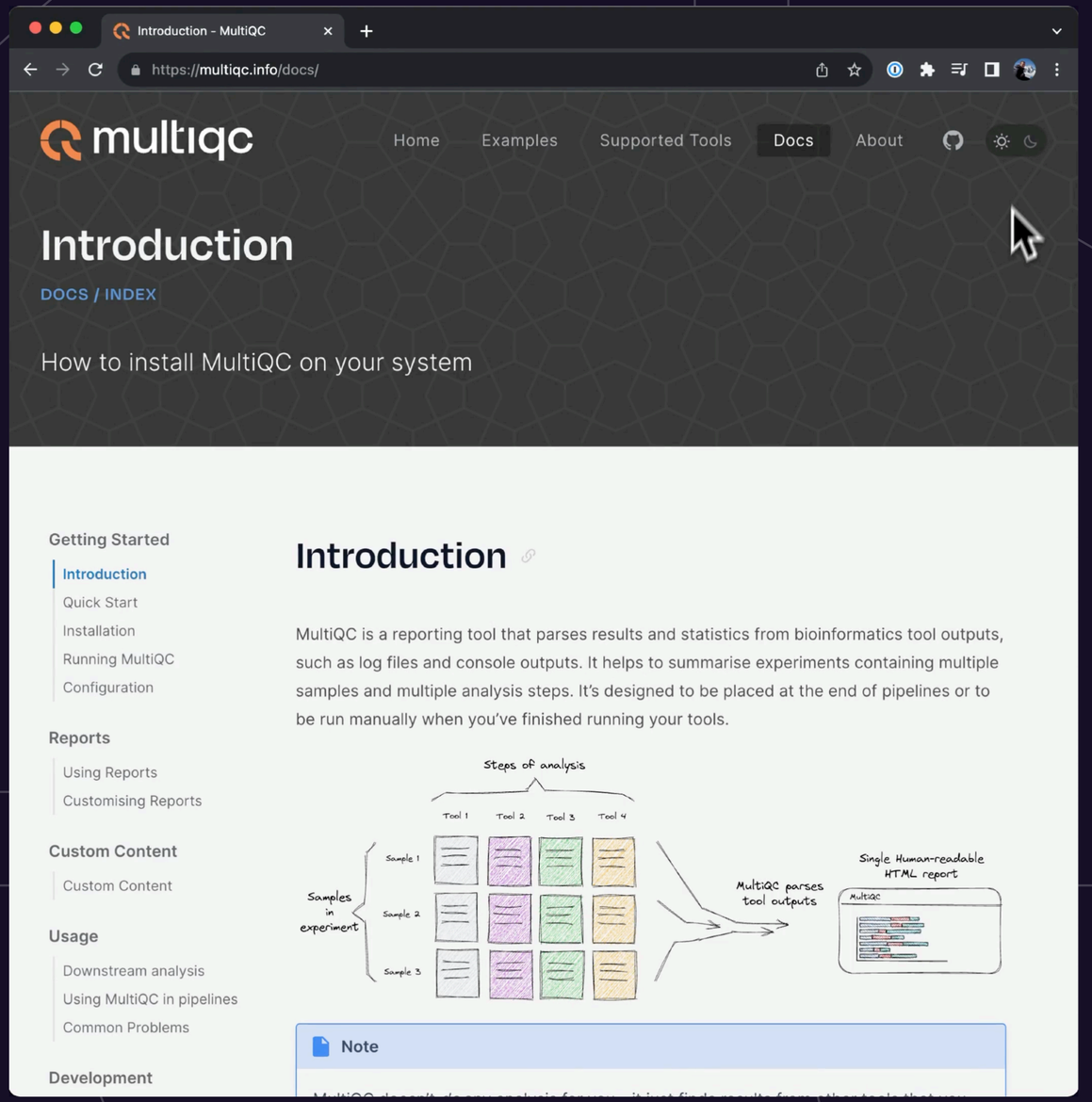
Dark mode

seqera

Updates to the default template

A fresh lick of pixels

- Dark mode 😎 🖤
- Updates frameworks
- *Bonus*: Find the easter egg on multiqc.info!



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

seqera

Standardised file format

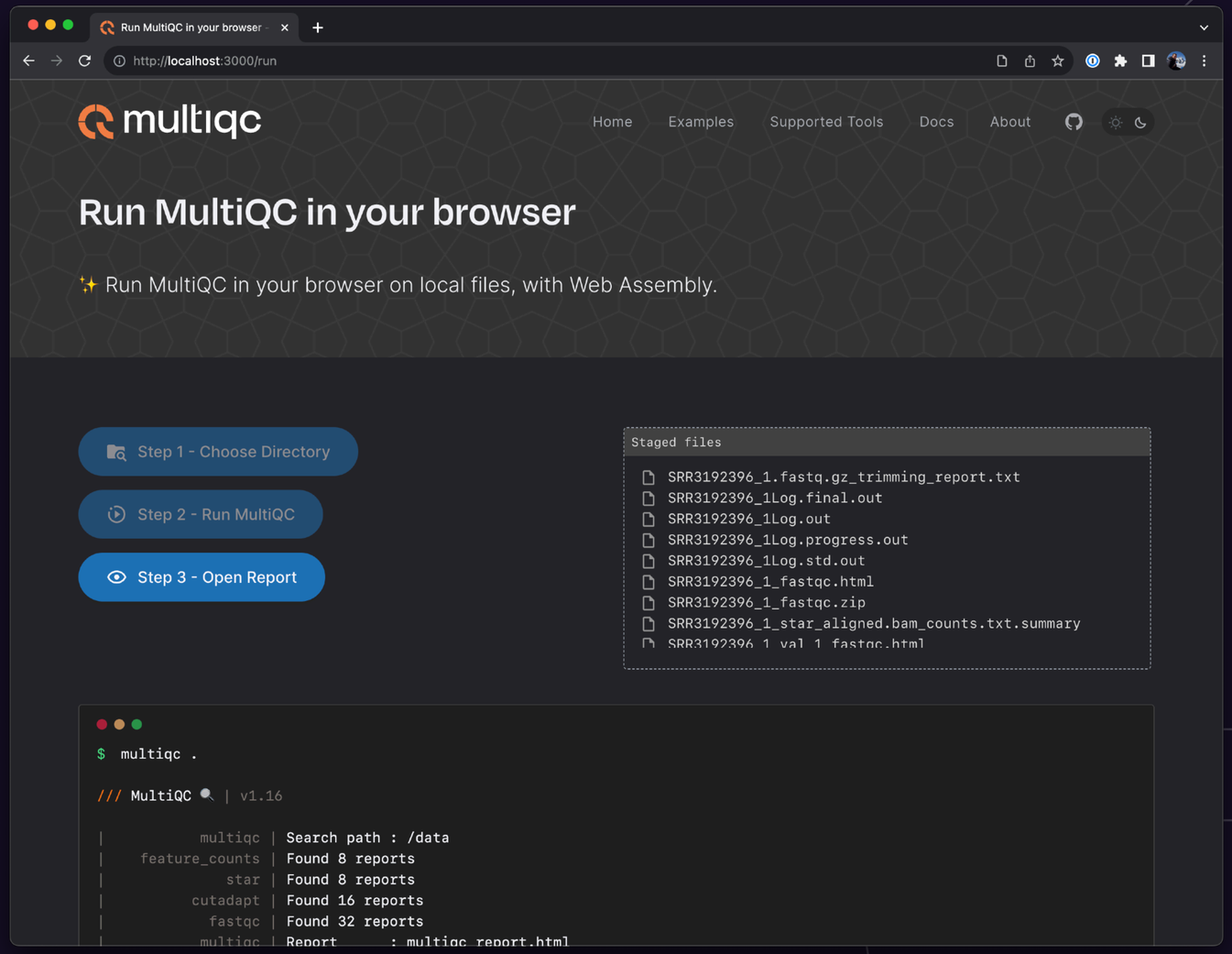
MultiQC as a foundation to build upon

- 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

seqera

Generate reports in your browser



multiqc.info/run

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

seqera

multiqc

Thank you

Phil Ewels, PhD

Product Manager for Open Source

phil.ewels@seqera.io