nextflow SUMMIT by seqera

Bioinformatics in the agentic age

Practical tips learnt from RustQC and rewrites.bio

Phil Ewels,
Senior Product Manager for Open Source, Seqera

Bioinformatics software, written by AI, is here.

Pre-AI

No software written with AI

AI-Transition

Old software re-written with AI

New software written with AI

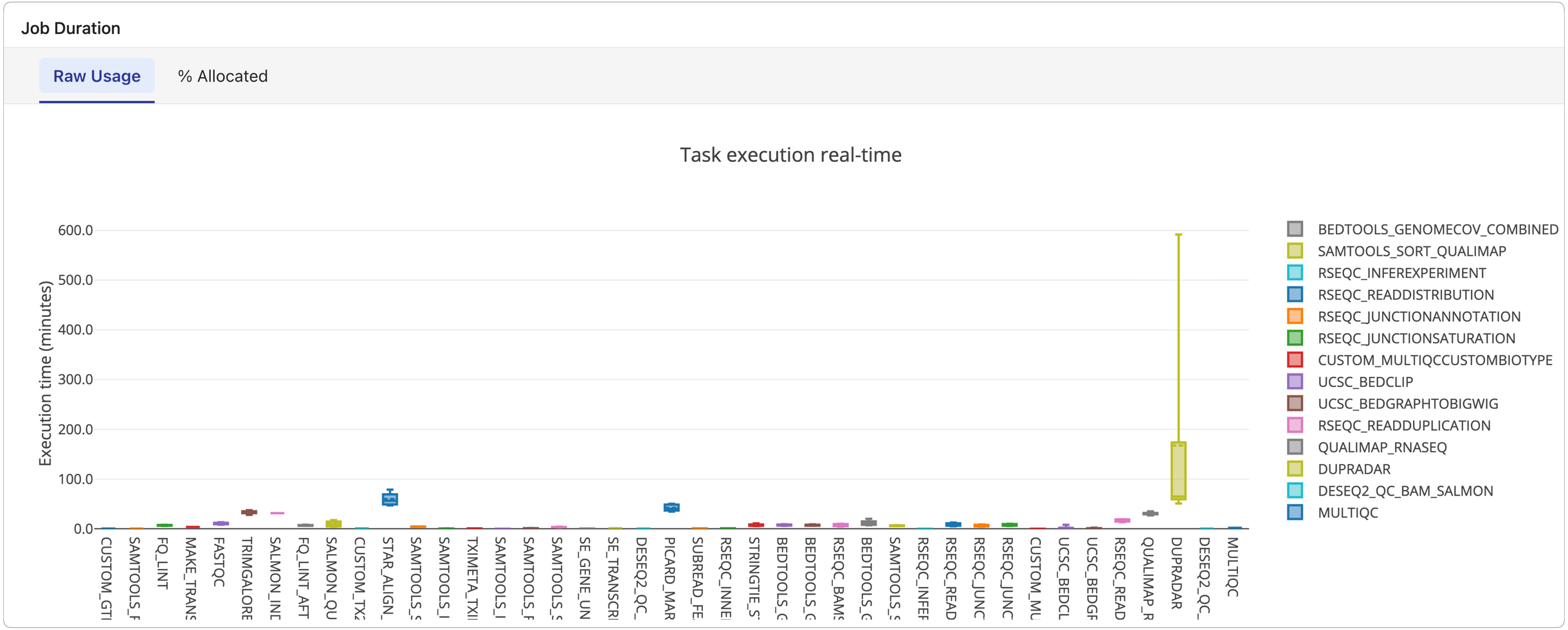
Post-AI

All software written with AI

01

RustQC

nf-core/rnaseq



nf-core/rnaseq

66h 12m total compute across 327 tasks

54 unique processes

** without Preseq & RSeQC TIN*

nf-core/rnaseq

66h 12m total compute across 327 tasks
54 unique processes

QC: 34h 24m (52%) Other: 31h 47m (48%)

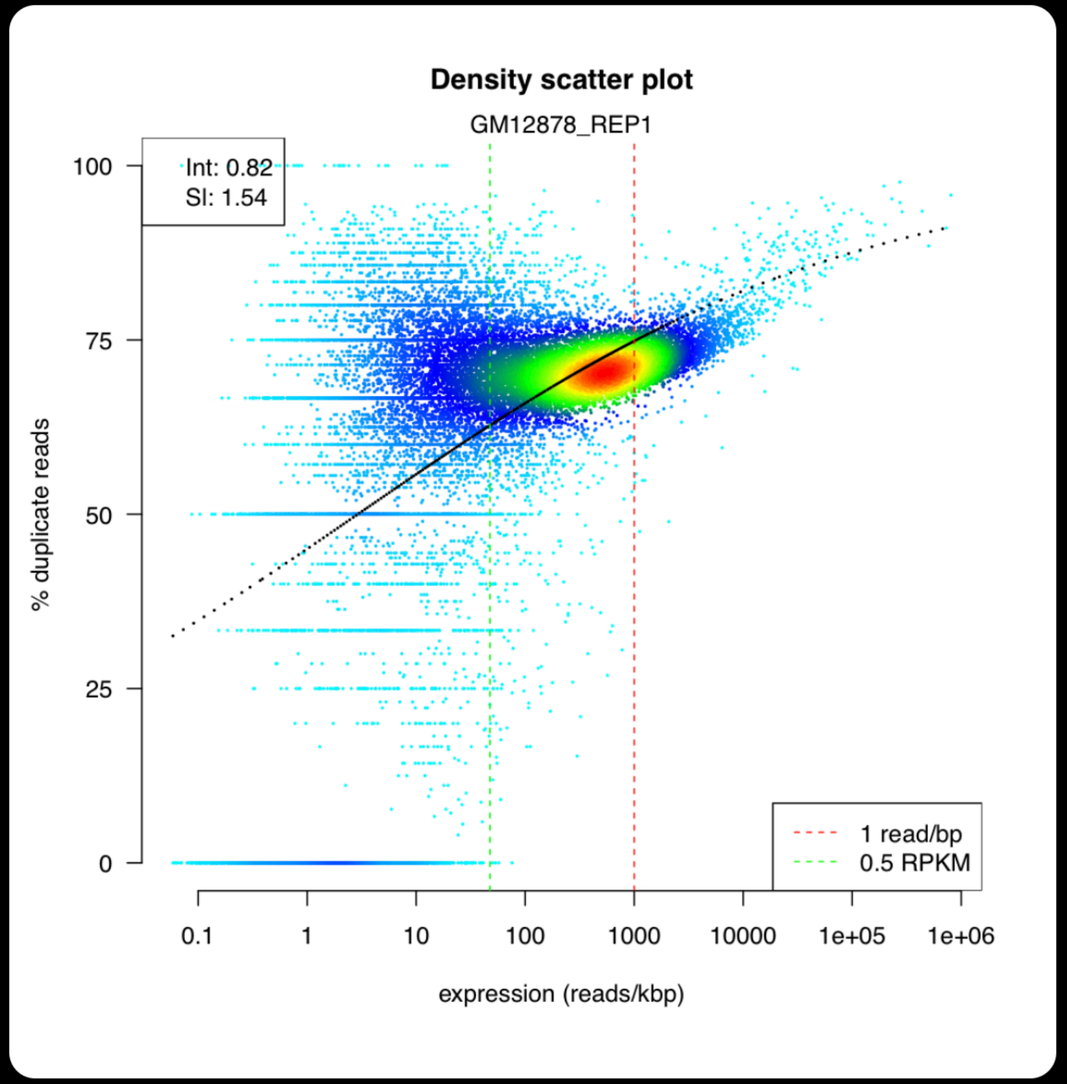
109h 18m total compute across 313 tasks
54 unique processes

QC: 78h 13m (72%) Other: 31h 4m (28%)

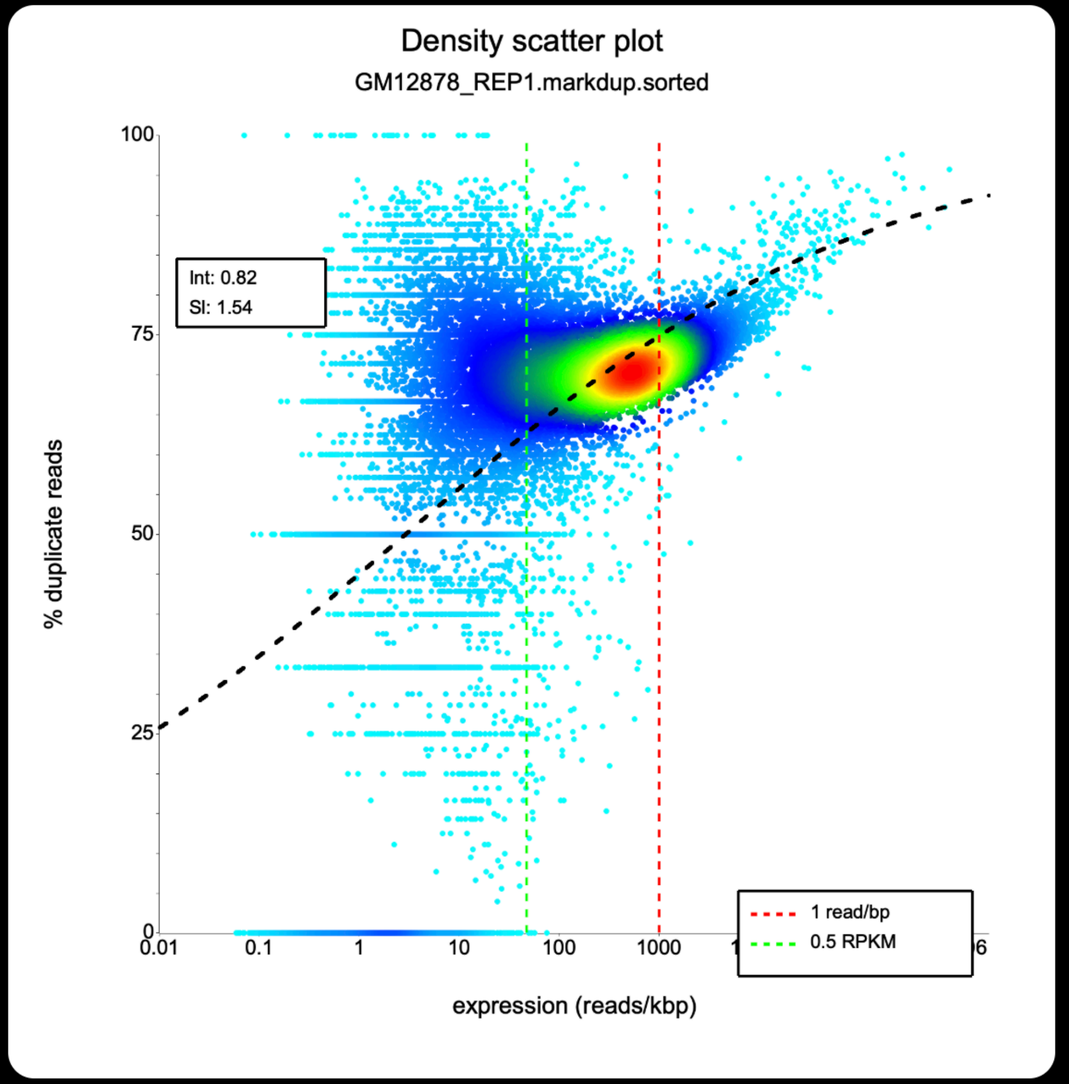
Phil Ewels 12:47 PM

I did a dupRadar Rust rewrite over lunch.. Testing it now 🧪

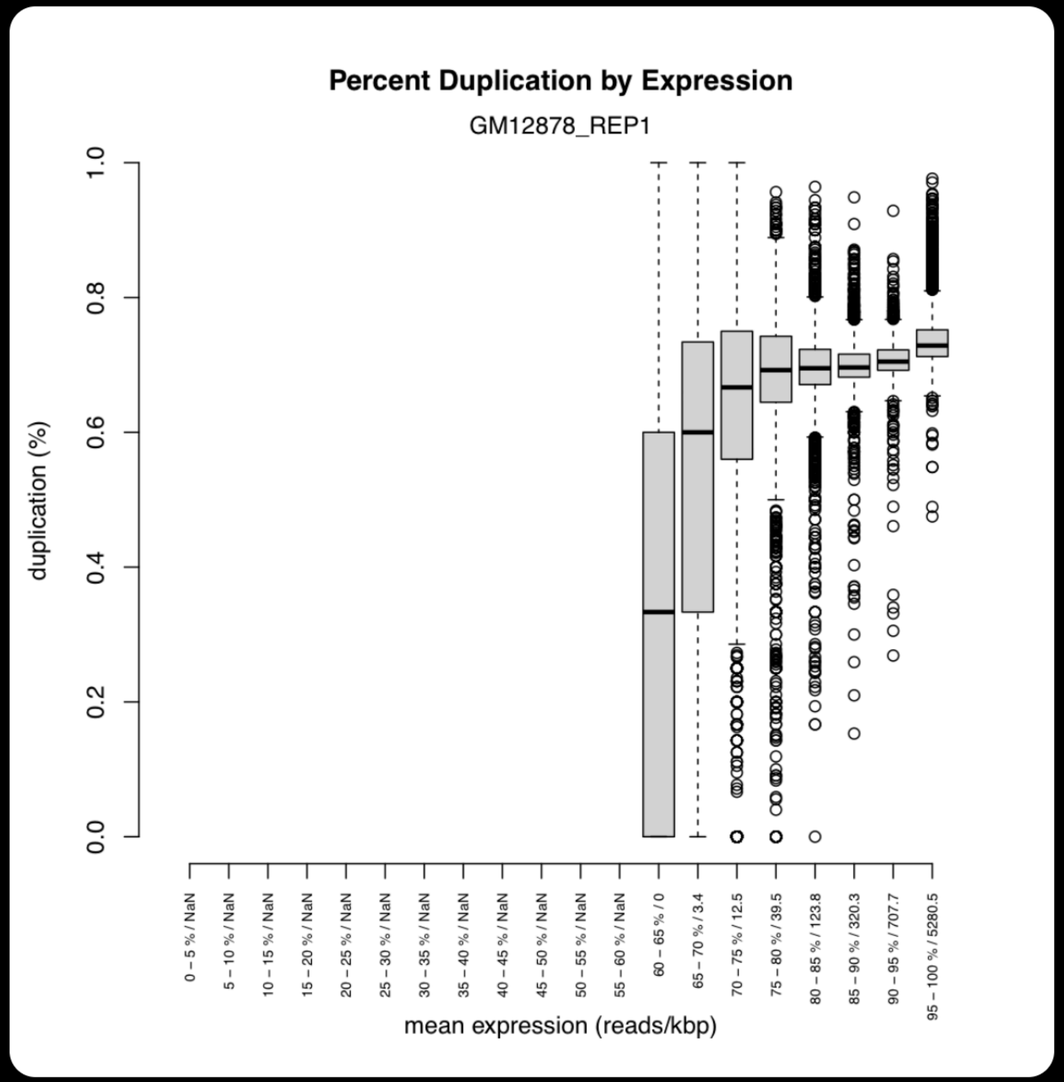
dupRadar



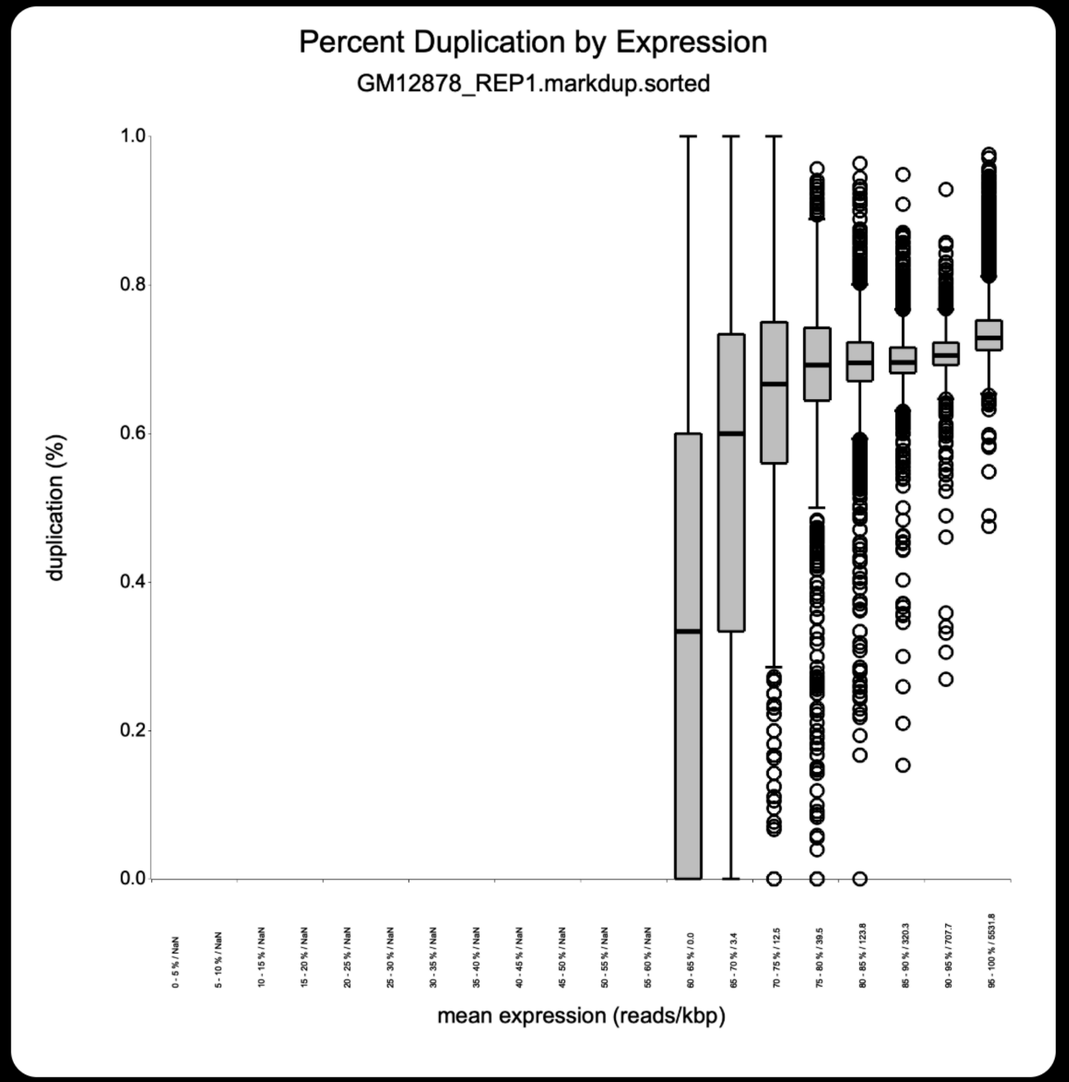
RustQC



dupRadar



RustQC



Phil Ewels 12:47 PM

I did a dupRadar Rust rewrite over lunch.. Testing it now 🧪

Phil Ewels 2:12 PM

I realise that I might have accidentally also made a super fast Rust implementation of featureCounts in the process of doing this 🙃

dupRadar (+featureCounts)

featureCounts (biotypes)

RSeQC - bam_stat
RSeQC - infer_experiment
RSeQC - read_duplication
RSeQC - read_distribution
RSeQC - junction_annotation
RSeQC - junction_saturation
RSeQC - inner_distance

Qualimap
~~(Samtools sort)~~

Samtools stats
Samtools idxstats
Samtools flagstat

Preseq

RSeQC - tin

16 commands into one

2.5 TB → 0.1 TB I/O

(for 8 samples)

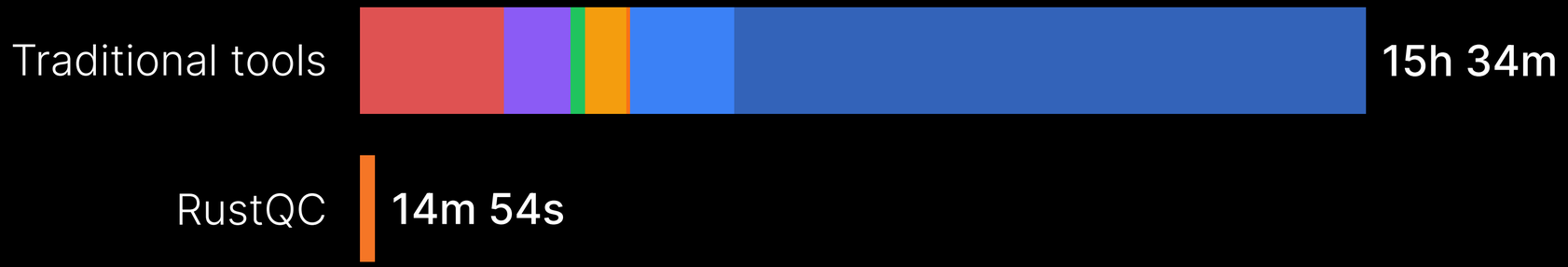
18 BAM file reads → 1

16 commands into one

60x faster

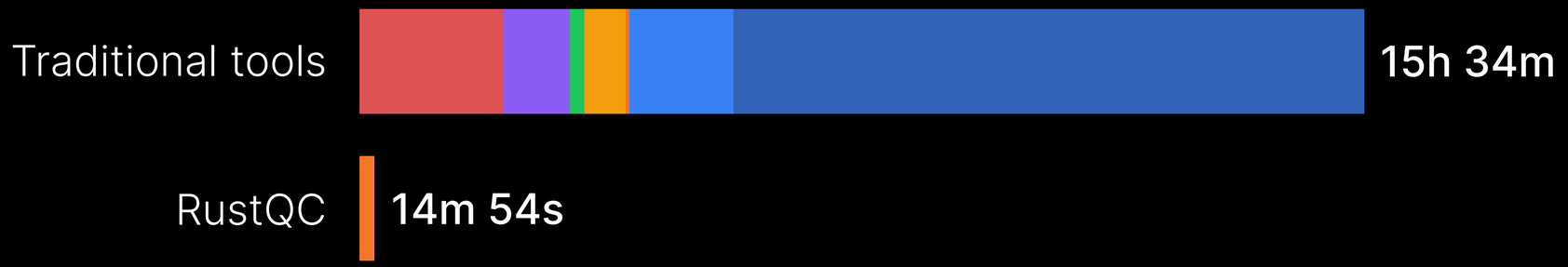
nf-core/rnaseq

QC sub workflow



nf-core/rnaseq

QC sub workflow

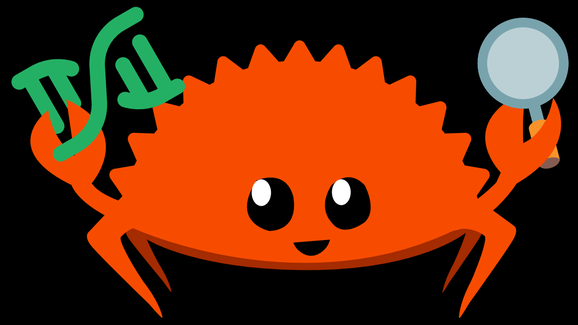


1.5M
CPU-hours saved

98%
less energy

~150t
CO_2e saved

For 100k samples/year



RustQC

02

Validation

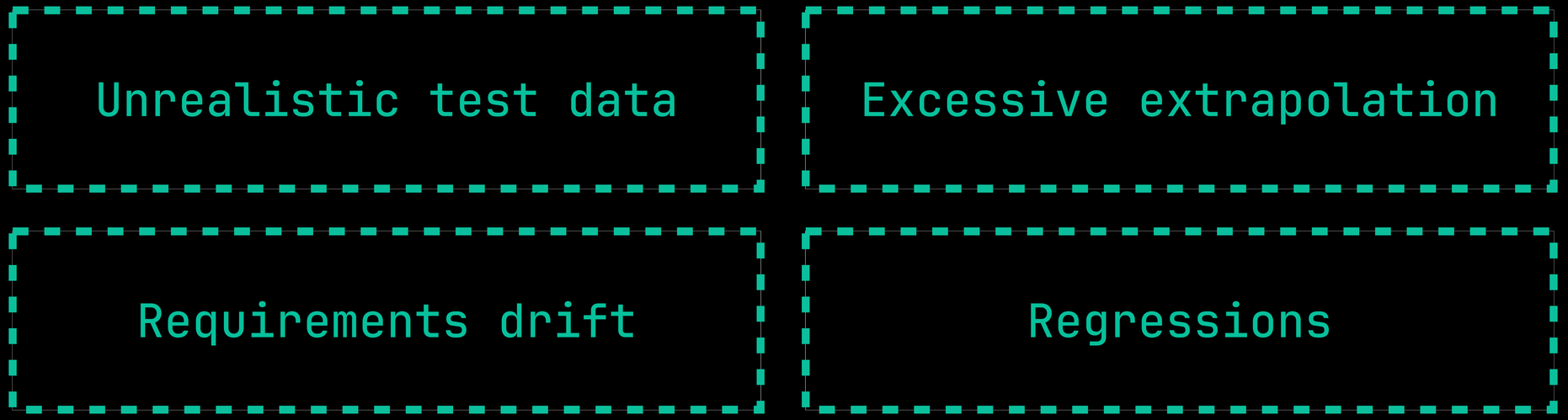
LLMs are built to please.

They're often wrong. ***Confidently*** wrong.

How can we tell?

Exact rewrites give us a specific result to aim for

Categories of LLM problems



Unrealistic test data

Excessive extrapolation

Requirements drift

Regressions

```
❯ Keep iterating until outputs are identical.
  My boss is watching.
```

```
❯ Keep iterating until nf-test passes 🚀
```

Snapshot testing of upstream tool outputs gives a verifiable target for agents

Snapshot testing of upstream tool outputs gives a verifiable target for agents

Nextflow is the perfect harness for autonomous, AI-driven rewrites

03

rewrites.bio

A MANIFESTO FOR BIOINFORMATICS

rewrites.bio

Rewriting bioinformatics tools with AI. Responsibly.

seqera

x

Principles for rewriting bioinformatics tools with AI

Philosophy

- Credit the original authors
- Emulate exactly
- Be transparent about AI

Planning

https://rewrites.bio

- Think big
- Work small

Building

- Test and benchmark with real data
- Build only what you need
- Pin versions and document

Stewardship

- Maintain and govern
- Preserve compatibility
- Release as open source
- Contribute upstream responsibly

General feedback

Matthias De Smet 📦 Apr 2nd at 12:42

You cannot believe HOW MUCH I LOVE THIS ❤️❤️

Dr Andrew Lonsdale @lonsbio.bsky.social · 18d

The almost best thing about this is that Bioinformatics Twitter is back!

▲ teekert 26 days ago | prev | next [–]

So needed. I just threw genebody_coverage and TIN out of RSeQC ;) Maybe they can come back now.

Nick Minor @nickminor.bsky.social · 27d

This looks amazing Phil! Let the Rust rewrites keep coming😎

Nils Homer 🌴 Apr 2nd at 14:39

Congrats @Phil Ewels, this is fantastic. 60x faster is hard to argue with. Really glad to see rewrites.bio too. We've been thinking about a lot of the same things with `fgumi`: credit the original work, match outputs exactly, validate obsessively.

▲ tenzin12 27 days ago | prev [–]

Really interesting direction. The validation and drop-in compatibility part is what caught my attention most. If this holds up across more datasets, this could remove a huge amount of RNA-seq QC runtime pain.

Mazdak Salavati · 1st 1w

Reader in Data Science at Dairy Research and Innovation Centre (SRUC)

Brilliant work Phill. Really happy to see your rewrites.bio approach to giving credit to original authors and preventing tool output fragmentation. Would be keen to see if we can ressurect old but critical tools in BioInf this way. Brilliant direction of travel.

Love · 1 | Reply · 2 replies

Frequent discussion points

Academic integrity

Why not improve tools?

Upstream contributions

AI clone army

04

Other rewrites

Other AI-assisted rewrites

SSHash + piscem
(Rob Patro) Sequence dictionary and read mapper

ruSTAR
(James Ferguson) STAR aligner

fgumi
(Fulcrum Genomics) UMI toolkit

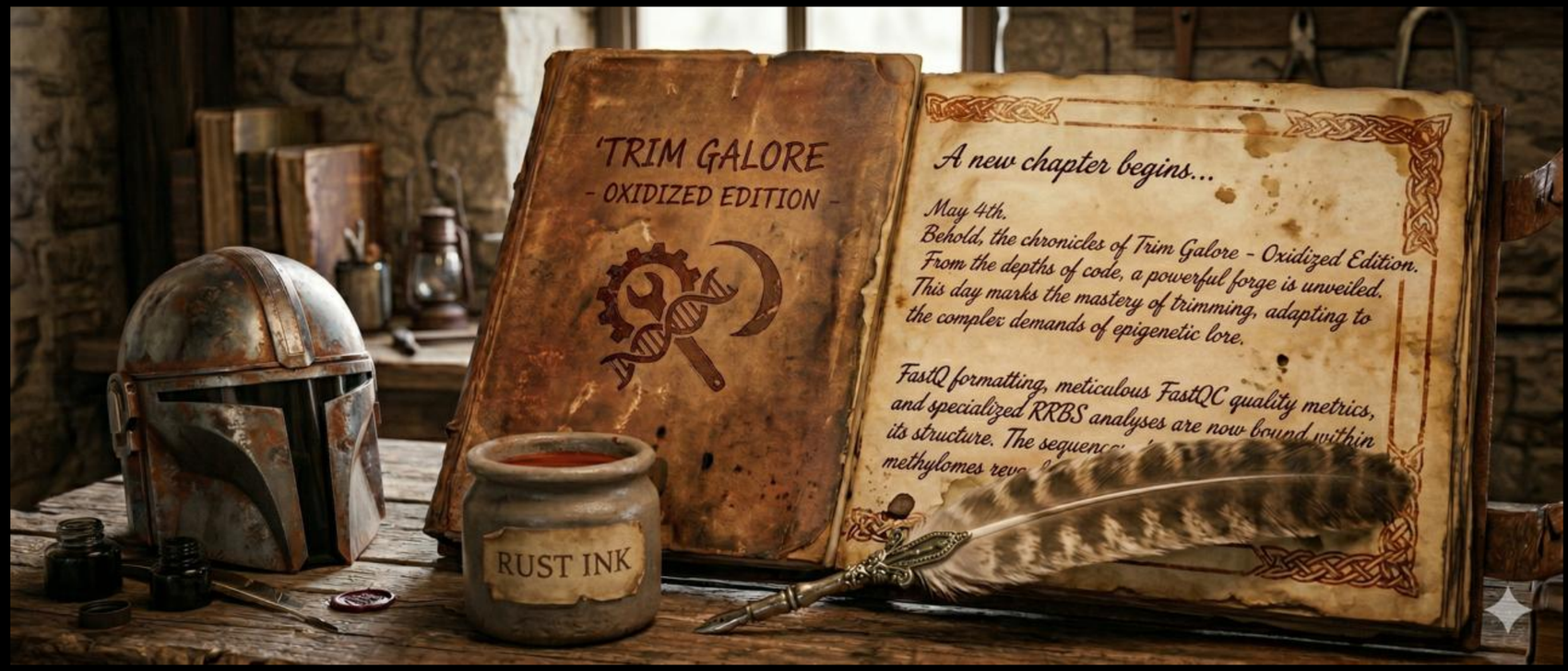
Trim Galore
(Felix Krueger) Read trimmer

Riker
(Fulcrum Genomics) Picard - QC

FastQC
(Phil Ewels) Sequence QC

..and many more

Trim Galore (Felix Krueger)



2.3 - 5x less CPU time

https://felixkrueger.github.io/TrimGalore/

Rewrites are an opportunity for new docs

Trim *Galore.*

https://felixkrueger.github.io/TrimGalore/

FastQC in Rust

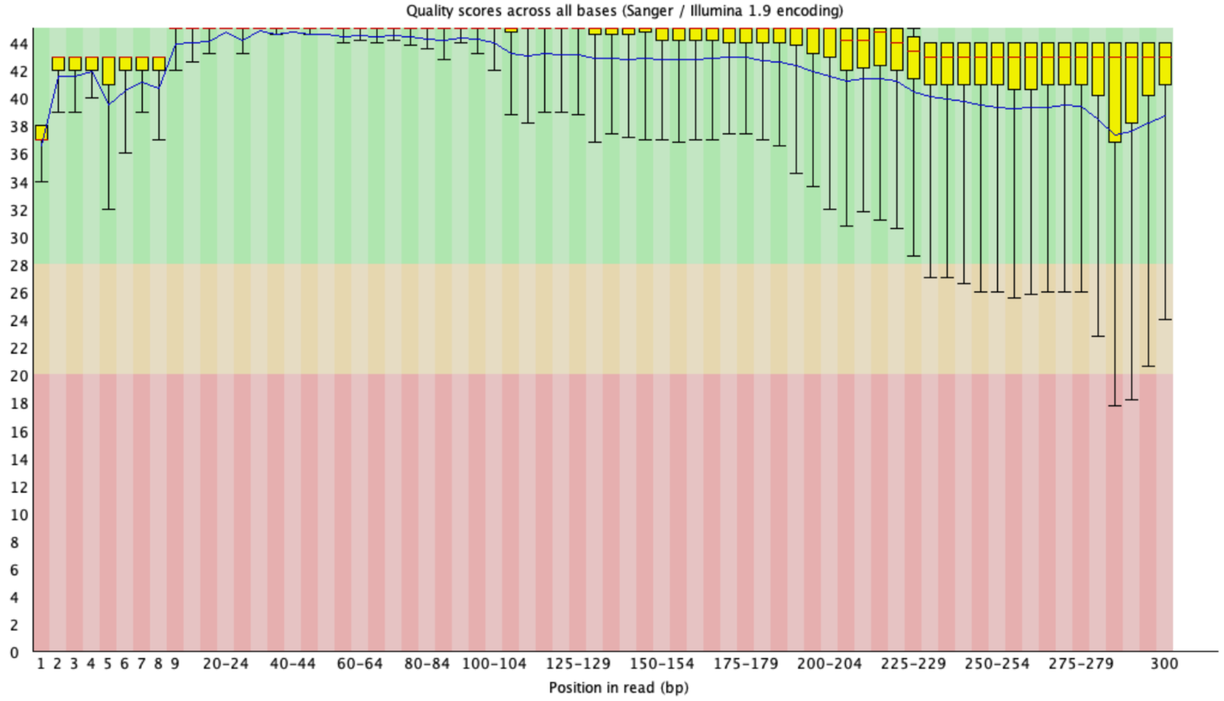
FastQC Report

Wed 8 Apr 2026
ERR16944282_1.fastq.gz

Summary

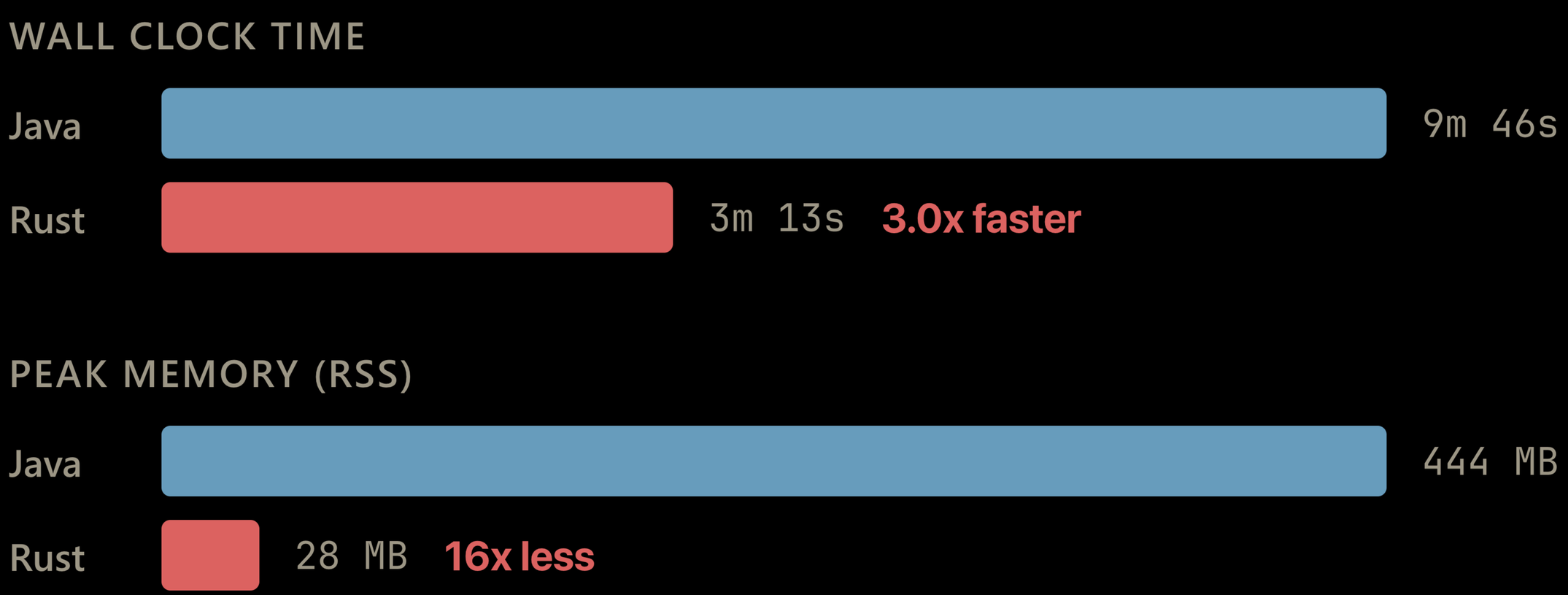
- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Per base sequence quality



Produced by FastQC (version 0.12.2.devel)

FastQC in Rust



https://ewels.github.io/FastQC-Rust/

FastQC in Rust: Why?

fastqc-rs

BioFastq

RastQC

falco

biobam

fastqp

fastp

https://ewels.github.io/FastQC-Rust/

FastQC in Rust: Why?

WHY

Test bed for development

Rust crate (library)

HOW

Track upstream closely

Contribute features upstream

https://ewels.github.io/FastQC-Rust/

x

FastQC in ~~Rust~~ Java: Upstream contributions

- #185 Bundled font
- #185 Trailing Adapter Content with `--min_length`
- #161 New report template, with design refresh
- #xxx 2-3x speed gain *(Paolo)*
- #xxx Stand-alone binary: no Java required *(Paolo)*

https://ewels.github.io/FastQC-Rust/

05

Looking ahead

Looking ahead

Nextflow as an agentic harness
Accurate validation is more important than ever

Gold-rush for rewrites of existing tools
An orphanage for abandoned bioinformatics tools

Accelerated adoption of Rust
Compatible library APIs to enable architectural rewrites

Community driven bioinformatics
Collaboration, validation and trust will be the driving force

> Empathy matters more than technical benefit or legal correctness.

Heng Li, Bioinformatics legend

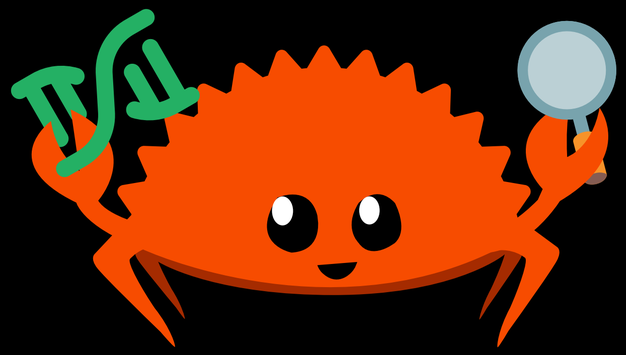
Blog post: "The AI Rewrite Dilemma"

https://lh3.github.io/2026/04/17/the-ai-rewrite-dilemma

Bioinformatics software, written by AI, is here.

Bioinformatics software, written by AI, is here.

Let's do it responsibly, together.



RustQC

https://seqeralabs.github.io/RustQC/

rewrites.bio