Open-Source Tools / Recent releases

Fulcrum builds and maintains open-source bioinformatics because the field works better when the infrastructure is shared, inspectable, and reproducible. All free, all in production pipelines, and we'd be glad to hear how they fit (or don't) into yours.

01 / NEXTFLOW · NF-CORE

fastquorum

UMI · PIPELINE

Reproducible UMI consensus pipeline.

Most UMI consensus workflows are bespoke glue scripts that drift between projects and break. fastquorum is an nf-core Nextflow pipeline implementing the fgbio FASTQ-to-consensus best-practice workflow with the same logic and outputs on a laptop, an HPC cluster, or in the cloud. A clean, versioned starting point for rare variant detection, duplex sequencing, or any workflow where UMI consensus is used.

nf-co.re/fastquorum

02 / RUST · ALPHA

fgumi

25× FASTER

A 25× faster Rust rewrite of fgbio's UMI tools.

fgbio is single-threaded by design, and runtimes climb steeply as panels deepen and the field moves from small targeted panels to exome- and genome-scale error- corrected sequencing. fgumi is a multi-threaded Rust reimplementation of the core UMI tools (extract, group, consensus, filter, metrics), intended as a drop-in equivalent. Our deepest benchmark runs in 70 seconds vs 30 minutes in fgbio.

github.com/fulcrumgenomics/fgumi

03 / R · BIOCONDUCTOR

MutSeqR

REGULATORY

Standardized ECS analysis for genetic toxicology.

Co-developed with Health Canada and the University of Ottawa, this R/Bioconductor package brings open, reproducible analysis to error-corrected sequencing data in a regulatory context. Variant filtering and classification, mutation frequency, dose-response and benchmark dose modeling aligned with Health Canada, EPA, and EFSA methods, plus mutation spectrum and signature tools.

github.com/EHSRB-BSRSE-Bioinformatics/MutSeqR

04 / RUST · QC

Riker

PICARD SUCCESSOR

Sequencing QC fast enough to run every time.

Picard has been the default for sequencing QC metrics for years (Tim Fennell and his team at Broad built it), but its performance hasn't scaled with current throughput. The result: teams trim QC for routine runs and reserve deep metrics for troubleshooting. Riker is a Rust-based successor with updated metric implementations and enough speed (10-15x faster than Picard) that deep QC on every dataset becomes the default.

github.com/fulcrumgenomics/riker

Bioinformatics expertise trusted by scientists who demand accuracy, reproducibility, and speed.

CODE	github.com/fulcrumgenomics
WRITEUPS	blog.fulcrumgenomics.com
CONSULTING	fulcrumgenomics.com

05 / CLI · WEB

ref-solver

REFERENCE ID

Identify which human reference a file was aligned to.

For when a collaborator hands you a BAM whose filename doesn't tell you which patch release it actually is, ref-solver compares the sequence dictionary — contig names, lengths, ordering — against a curated catalog covering UCSC, NCBI/Ensembl, Broad bundles, T2T-CHM13, and GRCh38 patches p1–p14. It never touches sequence data, so it's fast and safe for sensitive datasets.

whatsmygenome.fulcrumgenomics.com

conda install ref-solver

06 / RUST · CLINICAL

ferro-hgvs

~2.5M/SEC

Fast, comprehensive HGVS variant parsing for clinical genomics.

Existing HGVS string parsers handle ~20 variants/sec locally; ferro-hgvs processes ~2.5M/sec normalize via a zero-copy Rust parser, supports more of the HGVS spec — including intronic splice variants others can't normalize — and ships with infrastructure (ferro prepare, ferro-benchmark) that improves the whole ecosystem.

hgvs.acgt.bio   conda install ferro-hgvs

07 / RUST · FASTQ

chelae

ADAPTOR TRIMMING

Faster, more accurate short-read adapter trimming.

Adapter trimming is a routine preprocessing step that shows up in pipeline after pipeline, which means slow or inconsistent behavior gets paid for over and over. chelae is a multi-threaded Rust toolkit for trimming and filtering short-read FASTQ data, including poly-G trimming, adapter trimming, quality trimming, read-structure-based hard trimming and UMI extraction, length filtering, and fastp-compatible JSON reporting for MultiQC.

github.com/fulcrumgenomics/chelae

08 / C++ · ALIGNMENT

bwa-mem3

READ ALIGNMENT

Faster bwa-mem2 with broader architecture support.

bwa-mem3 is Fulcrum’s fork of bwa-mem2, focused on practical improvements to a tool many teams already know: faster alignment, quality-of-life updates, support for non-Intel architectures, and operational features like shared-memory reference indexes for multi-sample runs. It makes an existing workhorse better for the way sequencing data is actually processed now.

github.com/fulcrumgenomics/bwa-mem3