Metapipeline-DNA: A Comprehensive Germline & Somatic Genomics Nextflow Pipeline

Yash Patel[1,2,3*], Chenghao Zhu[1,2*], Takafumi N. Yamaguchi[1,2,3*], Nicholas K. Wang[1,2], Nicholas Wiltsie[1,2,3], Alfredo E. Gonzalez[1,2], Helena K. Winata[1,2], Nicole Zeltser[1,2,4], Yu Pan[1,2,3], Mohammed Faizal Eeman Mootor[1,2,3], Timothy Sanders[1,2,3], Cyriac Kandoth[1,2], Sorel T. Fitz-Gibbon[1,2,3], Julie Livingstone[1,2,4], Lydia Y. Liu[1,2,4], Benjamin Carlin[1,2,3], Aaron Holmes[1,2], Jieun Oh[1,2], John Sahrmann[1,2], Shu Tao[1,2,3], Stefan Eng[1,2], Rupert Hugh-White[1,2], Kiarod Pashminehazar[1,2], Andrew Park[1,2], Arpi Beshlikyan[1,2], Madison Jordan[1,2], Selina Wu[1,2], Mao Tian[1,2], Jaron Arbet[1,2], Beth Neilsen[1,2], Yuan Zhe Bugh[1,2], Gina Kim[1,2], Joseph Salmingo[1,2], Wenshu Zhang[1,2], Roni Haas[1,2], Aakarsh Anand[1,2], Edward Hwang[1,2], Anna Neiman-Golden[1,2], Philippa Steinberg[1,2], Wenyan Zhao[1,2], Prateek Anand[1,2], Brandon L. Tsai[1,2], Paul C. Boutros[1,2,3,4,5,§]

[1] Department of Human Genetics, University of California, Los Angeles, USA

[2] Jonsson Comprehensive Cancer Center, University of California, Los Angeles, USA

[3] Institute for Precision Health, University of California, Los Angeles, USA

[4] Department of Urology, University of California, Los Angeles, USA

[5] Broad Stem Cell Research Center, University of California, Los Angeles, USA

*These authors contributed equally to this work

[§]Corresponding author:
Dr. Paul C. Boutros
University of California Los Angeles
Los Angeles, California, 90095
Email: pboutros@mednet.ucla.edu
Phone: 310-794-7160

Abstract

Summary: DNA sequencing continues to get cheaper and faster. In parallel, algorithmic innovations have allowed inference of a wide range of nuclear, mitochondrial, somatic and evolutionary from DNA sequencing data. To make automated, high-quality DNA sequencing more readily available, we created an extensible Nextflow meta-pipeline called metapipeline-DNA. Metapipeline-DNA supports processing raw sequencing reads through alignment, variant detection, quality control and subclonal reconstruction. Each step supports quality-control, data-visualization and multiple algorithms. Metapipeline-DNA is cloud-compatible and highly configurable, with options to subsect, optimize and optimize analyses, including with automated failure-recovery. Metapipeline-DNA enables high-scale, fault-tolerant, comprehensive analysis of genome sequencing.

Availability: Metapipeline-DNA is an open-source Nextflow pipeline under the GPLv2 license and is available at https://github.com/uclahs-cds/metapipeline-DNA.

Introduction

High-throughput technologies have transformed biomedicine into a data-intensive field. DNA sequencing is one of key enabling technology, used in routine clinical care and a wide range of research studies[1]. Ongoing technological improvements in DNA sequencing continue to reduce costs and enable new discoveries, like the study of complex structural variants (SVs) and repetitive genomic regions by long-read sequencing[2]. Modern germline DNA sequencing studies routinely quantify single-nucleotide polymorphisms (SNPs), SVs, telomere length, mitochondrial variation and copy number[3,4], and many other features[5].

DNA sequencing has been especially helpful in improving our understanding of cancer. In typical tumour-sequencing studies both a sample of a cancer and a “reference” normal sample from the same individual are sequenced to better distinguish somatic from germline variation. Cancers are characterized by widespread genomic rearrangements, variation in mutation clonality, specific patterns of somatic mutations associated with carcinogens or other features, and a host of features absent or uncommon in germline sequencing like kataegis and chromothripsis[6]. Comprehensive analyses of cancer sequencing can improve diagnosis, prognosis and management[7,8].

The growing availability of DNA sequencing data has been paralleled by rapid development and adoption of both specific algorithms and workflow software. New discoveries often rely heavily on complex workflows comprising a mixture of established and novel algorithms[9]. These workflows, often termed “pipelines”, can be implemented in a range of orchestration frameworks like Galaxy[10], Snakemake[11], Common Workflow Language (CWL)[12], and Nextflow[13]. Workflows provide a way to automate processes by minimizing manual handling of data flow and facilitating stitching together of different tools to process raw data into refined forms.

The use of complex workflows has placed a growing emphasis on standardization, extensibility, quality control and compute infrastructure. Workflow implementations routinely differ across research groups, with many groups creating their own. These often lack key features like unit testing, integration testing, error-handling, fault-tolerance, input-output verification, quality-control, data-visualization and use of multiple algorithms to create consensus calls[14]. Given the volume of data and the expense of compute, workflows are often bespoke to the high-performance computing environment used by a single group[15]. Portability of workflows to new environments is part of the “model to data” (M2D) paradigm in data sharing and processing[16]. M2D overcomes the cost, time and privacy risks of data-transfer by bringing models or algorithms to the computing system where data is stored. M2D thus necessitates that models be portable across providers and environments to support workflow usage in conjunction with good data management principles hinging on findability, accessibility, interoperability, and reusability[17].

To address the need for a robust open-source DNA sequencing analysis pipeline, we created metapipeline-DNA. This Nextflow meta-pipeline is highly customizable and is capable of processing data from any stage of analysis. It can process DNA sequencing data starting from raw reads through alignment and recalibration, to variant calling, and even highly integrated analyses likely tumour subclonal reconstruction. Extensive quality control, testing and data-visualization are built into the workflow as a whole and into each individual step. It can work on multiple compute systems and clouds, facilitating analyses at any scale.

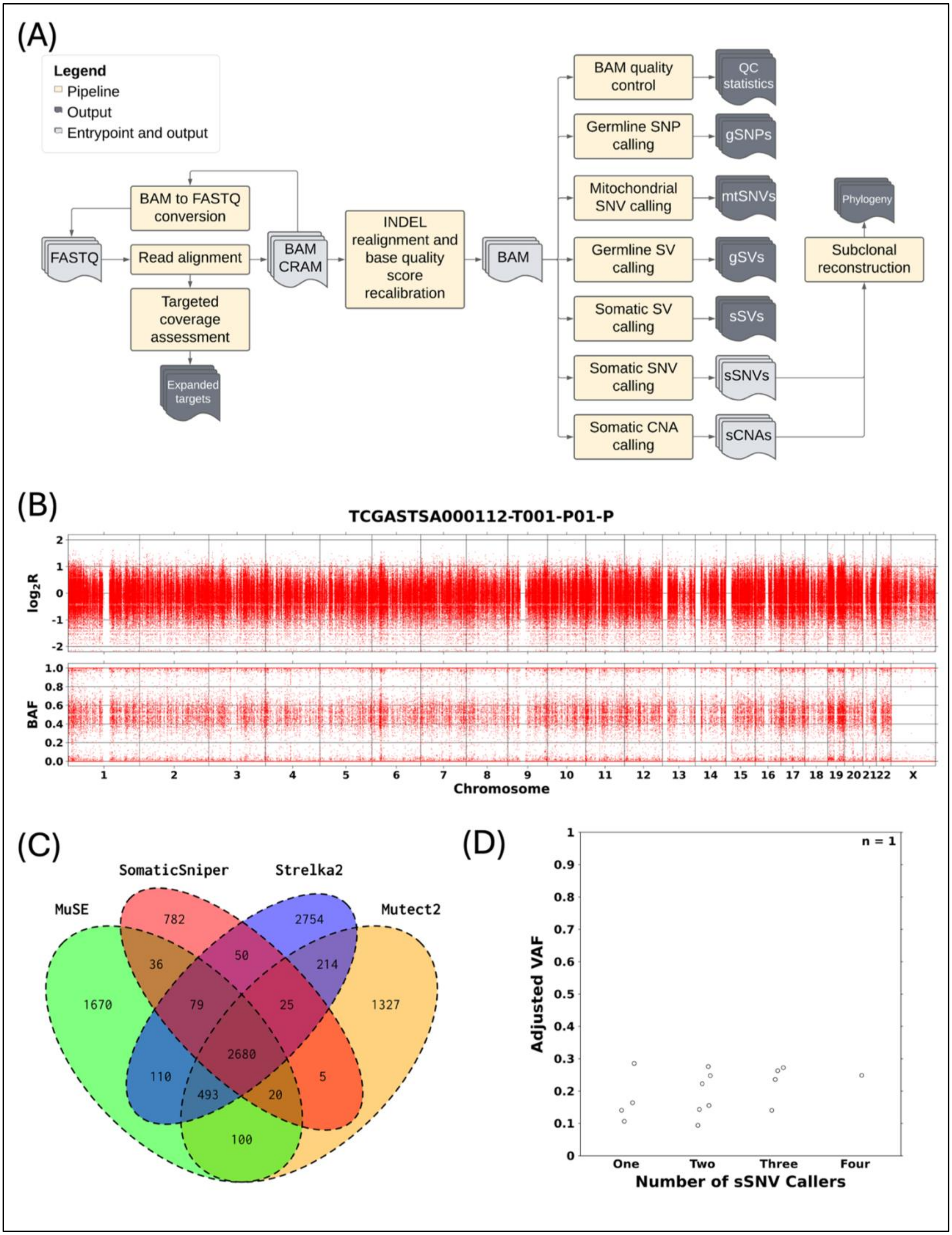

Figure 1. Data flow and visualizations. **A.** Data flow through metapipeline-DNA. **B.** Normalized tumour coverage relative to the matched normal ($\log_2$R) and the B-allele frequency of individual SNPs laid out across the genome to support CNA detection. **C.** Example intersection diagram of consensus variants between 4 SNV callers: MuSE2, SomaticSniper, Strelka2, and Mutect2. **D.** Variant allele frequencies based

on consensus between callers. VAFs are indicated for all combinations of consensus between one, two, three, and four variant callers, with each data point representing one combination. The adjusted VAF is calculated as an average of all variants present in the combination.

Results

Overview

Metapipeline-DNA is a Nextflow meta-pipeline for analysis of DNA sequencing data ranging from targeted sequencing to whole-genome sequencing. It encompasses 12 pipelines (**Table 1**) that collectively transform raw sequencing reads into sets of detected variants and other genetic and evolutionary features (**Figure 1A**). Most individual pipelines can execute multiple alternative algorithms and create consensus calls from them. For example, four separate algorithms can be executed for somatic single nucleotide variant (SNV) detection[14], automatically generating a consensus set of predictions and variant-associated data-visualizations (**Figure 1B-D**). Each pipeline can be executed independently and can be extensively parameterized to customize the selection and tuning of algorithms.

Several different sample run-modes are available, which we denote with the terminology *n*T-*m*N, where *n* indicates the number of tumour samples and *m* the number of reference samples (**Figure 2**). Thus, classic paired tumour-normal analysis is 1T-1N. Metapipeline-DNA fully supports modes like 0T-1N (*i.e.* germline DNA sequencing), 0T-3N (*e.g.* family trios), 1T-0N (*i.e.* unpaired tumour-only sequencing) and arbitrary multi-region tumour sequencing (*e.g.* 5T-1N). The primary limitation to multi-sample analyses is compute resource availability – particularly RAM and scratch-disk space. Metapipeline-DNA automatically handles input types for each mode and only executes feasible pipelines, independent of user-selections. For example, in 0T modes, variant detection is restricted to germline variants without users having to provide manual restrictions.

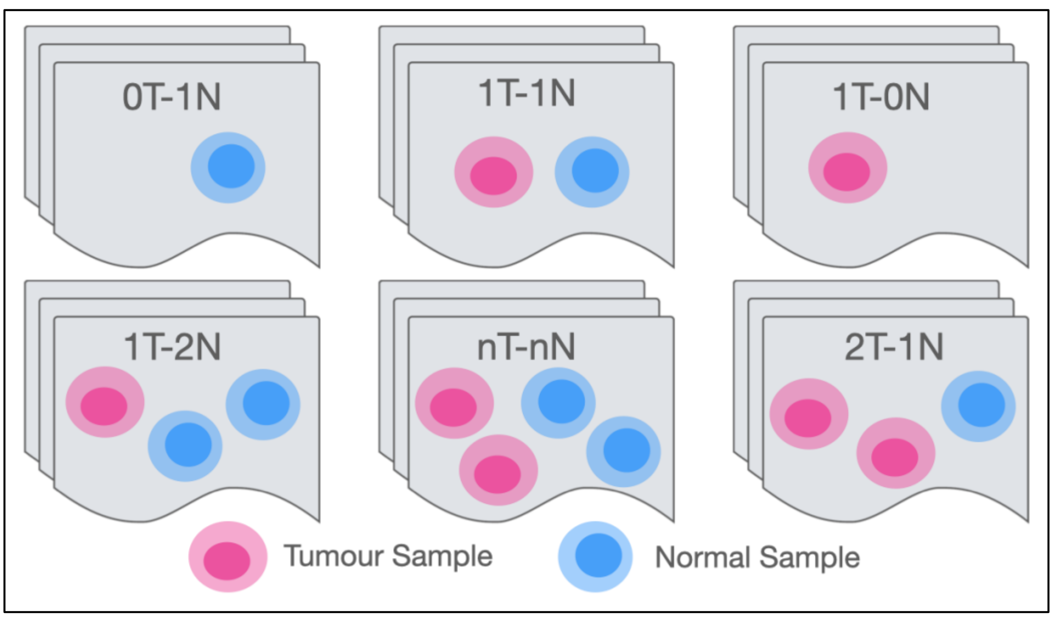

Figure 2. Runtime and peak physical memory usage per pipeline. Time and memory usage of pipelines per sample for the three different processing cohorts: PCAWG with GRCh38, PCAWG with GRCh37, and TCGA with GRCh38. Time is measured as wall-clock time taken by each pipeline and the total time taken by metapipeline-DNA. Memory is measured as the peak RAM usage by any single process by any pipeline.

The default mode of metapipeline-DNA accepts unaligned reads in FASTQ[18] format and executes all pipelines. A range of alternative entry-points are accepted, including aligned and unaligned BAM[19] and CRAM files, with automatic BAM-to-FASTQ conversions as needed. A few pipelines also accept alternative entry-points, such as SNV and copy number aberration (CNA) calls for tumour subclonal reconstruction[20] (**Figure 1A**). Documentation of all dependencies, input and output formats is available on standardized structured GitHub pages: current states at writing are summarized in **Supplementary Table 1**.

We engineered metapipeline-DNA to be intrinsically flexible with all necessary dependencies automatically identified and executed based on user selection. All run-modes and dependency identification have defaults set to the most common behaviour across thousands of runs, but with easy parameterization. For example, when input data is already aligned the default is to use these alignments. Nevertheless, configuration parameter allows the user to control whether reads are back-converted to FASTQ and re-aligned and whether aligned reads are recalibrated and so forth.

In a similar way, metapipeline-DNA is flexible to the specific genome build used, and has been tested extensively with GRCh37, GRCh38 and GRCm39. It can run in two basic modes: WGS mode and targeted-sequencing mode, based on user parameterization. Targeted-sequencing mode supports all subsets of the genome, including exome sequencing and arbitrary panels. Options are available to assess coverage, expand targets with off-target coverage sites, and automatically use expanded target intervals for downstream processing.

Pipeline	**Input Formats**	**Output Artefacts**	**Algorithms**	**Features**
Convert-BAM2FASTQ	BAM/CRAM	FASTQ	SAMtools	Automatic conversion from CRAM to BAM
Align-DNA	FASTQ	BAM	BWA-MEM2 HISAT2	Duplicate marking
Calculate-targeted-coverage	BAM Target region BED	Expanded regions Per-base depth in target regions and dbSNP sites Hybrid-selection metrics	SAMtools BEDtools	Automatic expansion of regions to off-target dbSNP loci with coverage
Recalibrate-BAM	BAM *Target regions*	INDEL realigned and base-quality score recalibrated BAM	GATK	Support for target regions Local INDEL realignment Base-quality score recalibration
Generate-SQC-BAM	BAM	BAM statistics Coverage metrics	SAMtools Picard Qualimap	Customizable selection of QC Coverage reporting and visualization
Call-gSNP	BAM *Target regions*	Per-sample GVCF Germline SNP VCF	GATK	Variant quality score recalibration Ambiguous variant filtration
Call-mtSNV	BAM/CRAM	Mitochondrial SNV VCF	MToolBox mitoCaller	Mitochondrial read extraction support for BAM and CRAM Heteroplasmy calling
Call-gSV	BAM	Germline SV VCF Germline SV BCF	DELLY Manta	Germline CNV calling Variant call QC
Call-sSV	BAM	Somatic SV VCF Somatic SV BCF	DELLY Manta	Germline SV filtration
Call-sSNV	BAM	Somatic SNV VCFs	Mutect2	Support for panel of normals

	Somatic SNV calls *Panel of normal*		Strelka2 SomaticSniper MuSE BCFtools-Intersect	Tumour-only mode Multi-tumour mode Consensus callset and visualization Variant allele frequency distribution by callset
Call-sCNA	BAM	Somatic CNA VCF or TSV	Battenberg FACETS	Standardized visualization Option for customizing Battenberg refit suggestions
Call-SRC	SNV calls CNA calls	SNV clustering Reconstructed phylogeny	PyClone PyClone-VI PhyloWGS DPClust FastClone CliP CONIPHER	Customizable combinations of clustering algorithm and phylogeny algorithm Standardized clustering and phylogeny formats

Table 1: metapipeline-DNA Constituent Pipelines. Pipelines encompassed within metapipeline-DNA and their inputs, outputs, algorithms, and key features. Inputs that are *italicized* are optional and inputs separated by "/" represent a list of choices from which one must be chosen.

Data Visualization & Quality-Control

Metapipeline-DNA includes a range of quality control steps and pipelines to assess data quality at many levels, including reads, alignments and variant calls. The reversion of alignment includes generation of SAM flag and alignment statistics of the original BAM along with a calculation and comparison of total reads before and after conversion to FASTQ to ensure no loss of reads. These quality-control analyses produce a variety of data-visualizations and reports. For example, alignment quality is inferred from BAM (or CRAM) files in a range of ways including coverage distributions over the genome (or target region with or without padding; **Figure 3A-B**). Reads are quantified by a range of quality metrics, including total counts, mapping qualities, GC content, insert sizes, read lengths, duplications and others. **Figure 3C** shows an example of read number stratified by a range of quality groupings. A range of software are used to generate these metrics, including SAMtools[19], Picard[21], and Qualimap[22]. Pileup summaries at common sites are generated and used as a precursor to estimate contamination across samples. In targeted-sequencing mode, additional coverage assessment is performed through per-base read depth calculations at target regions and well-characterized off-target polymorphic sites provided from dbSNP[23].

At the variant call level, variant-specific metrics are calculated, assessed, and visualized (**Figure 1B-D**). Germline SNP calls undergo filtration using models built from variant quality scores for both SNPs and Indels. Somatic SNVs are assessed based on consensus between callers and associated variant allele frequencies. Somatic CNAs are supported through genome-wide plots of normalized tumour coverage relative to the matched normal and B-allele frequencies (BAF).

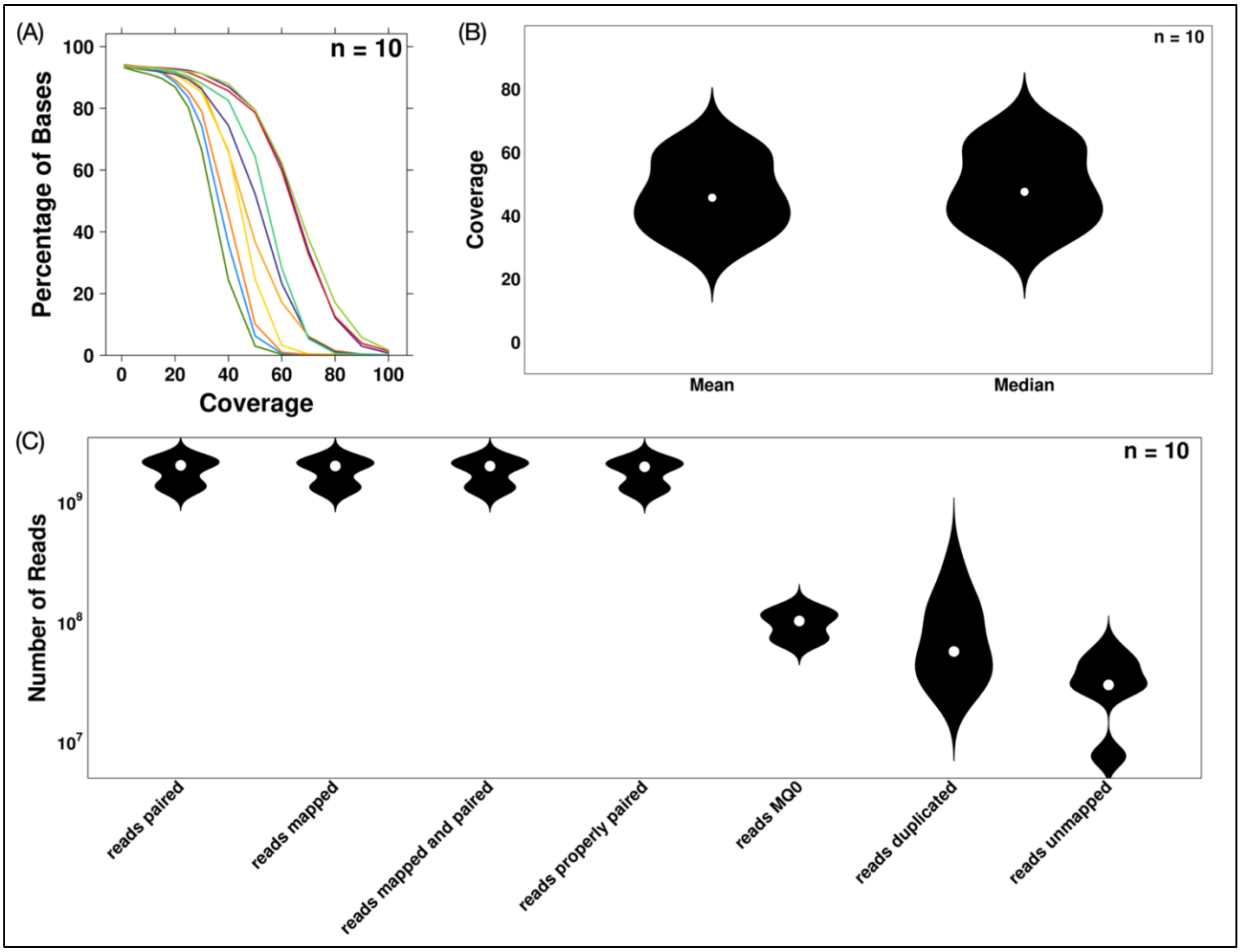

Figure 3: Alignment and coverage metrics. A. Percent of bases in the genome at each fold of coverage for normal and tumour samples (each line represents one sample) for all five WGS PCAWG patients. Each line represents a different sample, with the percentage of bases calculated using the coverage metrics. **B**. Distribution of mean and median coverage across all samples, highlighting two rough separations arising from the normal samples and tumour samples with the normal samples being in the lower coverage separation. **C.** Distributions of read numbers across alignment metrics including mapped/unmapped, low mapping quality, duplication, and paired.

Software-Engineering & Pipeline Robustness

We placed a heavy focus on generating re-usable and extensible software that could automatically detect and recover from common errors, particularly in the compute environment. This led us to adopt or create a series of development practices and pipeline features aimed at maximizing quality. All software is open-source, available on GitHub (https://github.com/uclahs-cds/metapipeline-DNA), with transparent tracking of issues and discussions. Development followed a test-driven approach using the NFTest framework[14]. Metapipeline-DNA has a suite of 71 unit, integration, and regression tests that are run for each new release with testing performed for different stages of execution from end-to-end tests to individual pipeline tests. Our extensive use of Docker containers allows seamless co-existence of multiple pipeline versions, and the combination of automated testing and containerization facilitates rapid updating with new features or dependency versions. Standardized GitHub issue templates support robust reporting of both bugs and new feature-requests, allowing ideal collaboration (**Supplementary Figure 1**). At writing, development has involved 42 contributors making 1,293 pull-requests, and 45 individuals making 995 suggestions, feature-requests and issue-reports across 13 pipeline repositories.

Bioinformatics data has high intrinsic variability, and bioinformatics software can be prone to significant numbers of failures – particularly in heterogeneous high-performance computing (HPC) environments. Failure handling is built into metapipeline-DNA to predict and minimize wasted computation. We automated input and parameter validation to catch issues prior to commitment of compute resources[24]. Validation of pipeline parameters is also implemented to foresee potential errors prior to resource commitment. Individual pipelines are modularized and set up to be fault-tolerant such that errors or failures in one pipeline stay isolated from and do not terminate other pipelines that are not their direct dependencies. With the robust input formats and configurable pipeline selection, metapipeline-DNA can be easily re-run in cases of failure, starting from prior partial results.

All outputs are organized with standardized directory and naming structures (**Supplementary Figure 2**). Filenames have been standardized to provide dataset, patient and sample information in a consistent way across pipelines. Metapipeline-DNA similarly organizes log-files to ensure saving of and ready access to the metapipeline-DNA logs, individual pipeline-level logs and compute partition logs. These logs capture execution and resource usage metrics for every process. Robust tooling has been developed around process and pipeline execution to ensure logs are captured for both successful and failing steps to enable debugging and record-keeping. Scripts have been created that automatically "crawl" over a series of pipeline runs to extract and tabulate information about run success, compute resources and other features.

Compute Infrastructure

Metapipeline-DNA includes customizability for compute infrastructure, execution, and scheduling in a distributed, cloud-agnostic workflow, with successful testing and validation performed in both Microsoft Azure and AWS computing environments. Execution follows the pattern of a single leading job responsible for submission and monitoring of per-sample or per-patient analysis jobs. Execution is currently performed with the Slurm executor with optional specification of compute partitions[25]. Parameters also exist to control rate of job submission and amount of parallelization/resources usage. Once configured and submitted, metapipeline-DNA automatically handles processing of an entire cohort with input parsing and job submission without user intervention. Real-time monitoring is available through email notifications sent from a server watching individual step start, end, and status. The choice of executor itself is parameterized and can be easily extended to other environments.

Metapipeline-DNA includes optimizations for disk usage, including (optional) eager intermediate file removal and built in checks to allow for optimized disk usage (performing I/O operations from high-performance working disks). Resource allocation for individual steps is also automatically handled, with pipelines running in parallel as available resources allows. Resource-related robustness is also built into pipelines to detect shortages in memory allocation and automatically retry processes with higher allocations.

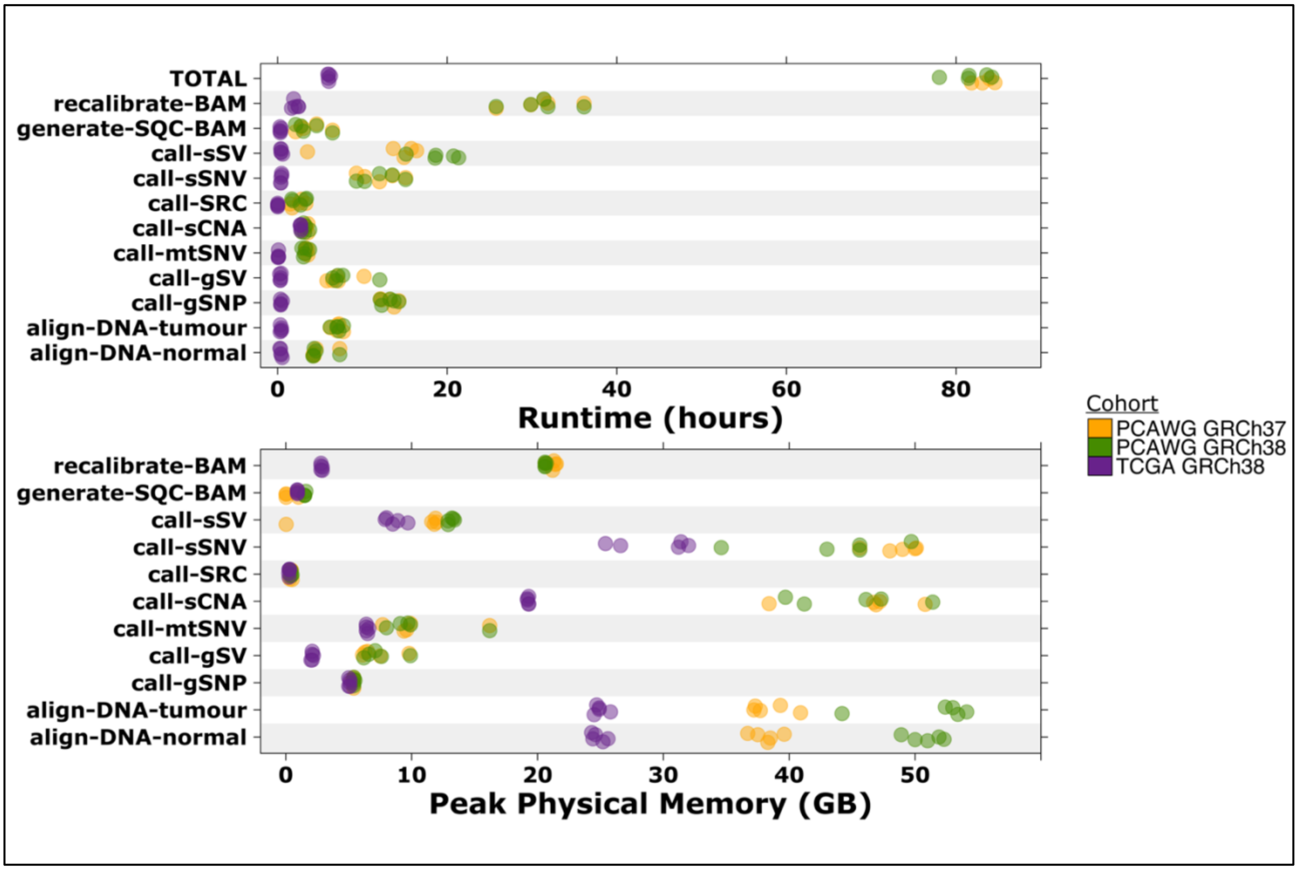

Figure 4: Subclonal reconstruction. Reconstructed phylogeny of tumour samples SA478344, SA528788, and SA528876 using consensus SNV callset comprising variants called by at least two out of four SNV callers (MuSE2, SomaticSniper, Strelka2, Mutect2) and FACETS CNAs. Nodes represent identified subclones with the evolutionary history depicted over SNV accumulation. Along the x-axis is the cellular prevalence (CP), indicating the fraction of all cells comprising each subclone.

Use-Case: PCAWG-63 normal-tumour pairs and TCGA sarcoma normal-tumour pairs

As a demonstration, ten normal-tumour pairs were processed through the entirety of metapipeline-DNA. Five pairs were selected from the Pan-Cancer Analysis of Whole Genomes (PCAWG)[26] 63 dataset and another five from The Cancer Genome Atlas (TCGA)[27]. The PCAWG-63 samples were sequenced with whole-genome sequencing and derived from multiple cancer types: one from uterine corpus endometrial carcinoma, one from biliary tract carcinoma, and three from esophageal adenocarcinoma. The samples had a median coverage of 63X (range: 45X-65X) for the tumour samples and 38X (range: 34X-54X) for the normal samples. The TCGA samples were derived from soft tissue sarcoma samples sequenced with exome-targeted sequencing. Both pairs were processed using metapipeline-DNA from alignment to subclonal reconstruction. The PCAWG-63 samples were processed with both GRCh38 and GRCh37, with similar runtimes across the two reference builds at an average of 81.76 hours (95% CI: ± 14.23) for GRCh38 and 83.36 hours (95% CI: ± 12.99) for GRCh37. Across the ten pairs, memory usage peaked in call-sSNV (average ± 95% CI: 48.54GB ± 2.30 and 29.32GB ± 3.82 for PCAWG63 GRCh37 and TCGA GRCh38 respectively) and in align-DNA (average ± 95% CI: 51.42 ± 5.07 for PCAWG63 GRCh38). Runtimes and peak memory usage of metapipeline-DNA for these samples are visualized in **Figure 4** and summarized in **Supplementary Table 2**. Examples of reconstructed phylogeny for three samples are shown in **Figure 5**. Variant allele frequencies aggregated over all combinations of consensus calls are shown for all samples in **Supplementary Figure 3**.

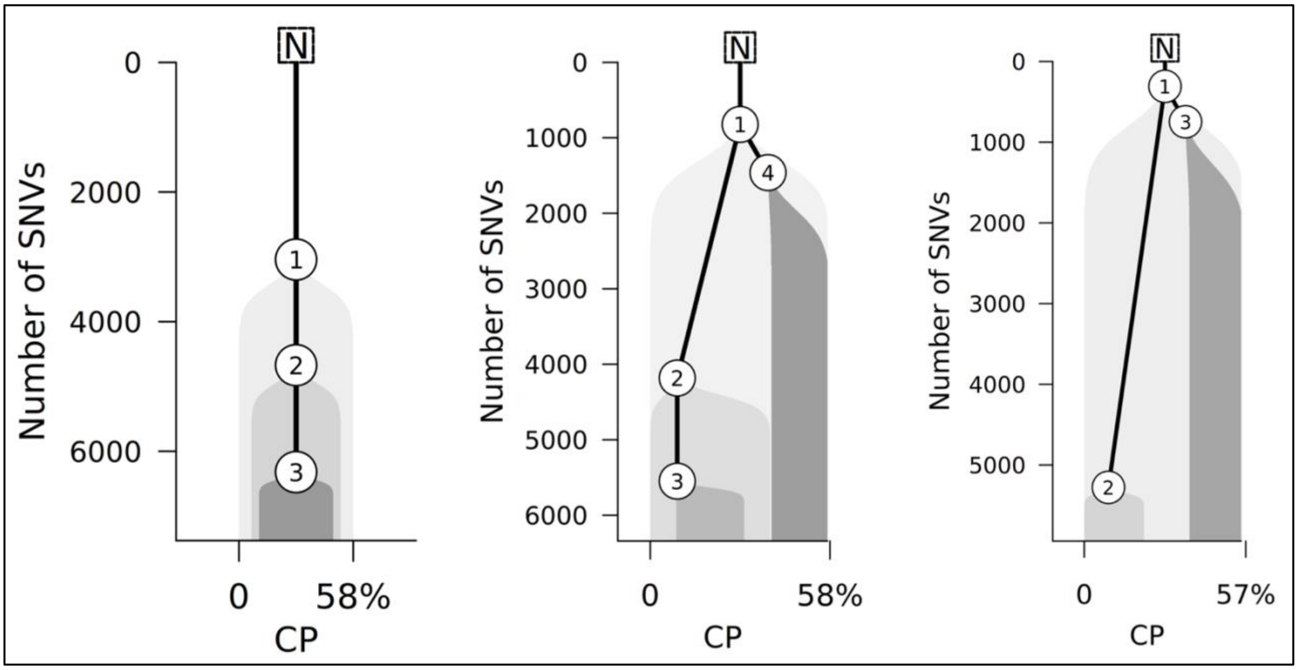

Figure 5: Metapipeline-DNA sample run-modes. Sample combinations supported by metapipeline-DNA. The nT-nN combination indicates any arbitrary numbers of normal and tumour samples. Each combination is automatically detected and considered during processing of all pipelines to select appropriate algorithms and processing modalities.

Discussion

Metapipeline-DNA was designed to facilitate the analysis of DNA sequencing data at scale while retaining the configurability and flexibility needed in academic environments. This is a key contrast to field programmable gate array (FPGA)-like approaches such as DRAGEN[28]. These strategies can have benefits in speed through hyper-optimization of hardware to suit specific algorithms and data formats but come at the expense of flexibility in areas with rapid ongoing methodologic development such as novel algorithms and sequencing technologies. As the field of genomics evolves, the ability to quickly integrate and test emerging methods becomes increasingly important, highlighting a limitation of fixed-function hardware solutions.

Metapipeline-DNA fills this key niche of supporting the rapidly expanding volume of sequencing data, supporting a range of existing tools and algorithms, and remaining flexible for ongoing expansion. By easing and optimizing the multi-step analyses intrinsic to DNA sequencing data, it reduces the barrier to incorporating new methods and analyzing large datasets. Indeed, it is entirely feasible for Metapipeline-DNA to leverage and incorporate FPGA-enabled and graphics processing units (GPU)-accelerated methods directly as part of its modular structure (e.g., for alignment); this is a key area of ongoing development.

Individual pipelines within metapipeline-DNA are organized in a modular fashion, allowing for a plug-and-play architecture that can be adapted to support additional technologies as they become available. Algorithms and workflows for processing long-read data, for example, pose an avenue for expanding the meta-pipeline as such tools mature and long-read datasets become more common. The context of DNA also brings up the possibility of similar meta-pipelines for other biological molecules such as RNA and proteins. Workflows across different biomolecules can share the architecture, automation and quality-control of metapipeline-DNA in a way that allows improvements to any one to improve others. Such workflows are currently under development to provide a similar level of configurability and extensibility for analyses of RNA and protein data.

The volume of data and size of individual samples being generated and processed in sequencing studies is often very large. With that comes a need for optimization of analysis pipelines' data handling. Metapipeline-DNA contains several disk usage optimizations to efficiently handle large amounts of data while minimizing I/O operations and cross-file system data movement. The framework connecting analyses automatically identifies necessary outputs from dependent pipelines and makes it available without any redundant copying or duplication. There are additional enhancements that are underway to minimize duplicated data and disk usage of metapipeline-DNA by building plugins to enable moving of files rather than copying when possible and optimizing individual pipelines to avoid shuffling around large output files.

Metapipeline-DNA is a highly customizable DNA sequencing analysis pipeline combining speed and flexibility in a modular framework to enable processing of data at any point from read alignment to tumour subclonal reconstruction. By facilitating the integration of diverse tools and supporting the rapid development of new methodologies, it positions itself as a versatile platform for future enhancements as novel DNA sequencing and analysis methods are developed.

Methods

Analysis Cohort

To demonstrate the use of metapipeline-DNA, we chose ten normal-tumour pairs. Five were WGS pairs from PCAWG-63: one from uterine corpus endometrial carcinoma donor DO43506, one from biliary tract carcinoma donor DO218695, and three from esophageal adenocarcinoma donors DO50342, DO50407, and DO50311. Five were exome sequencing pairs of soft tissue sarcoma pairs from TCGA donors TCGA-QQ-A8VD, TCGA-X6-A8C6, TCGA-HS-A5N8, TCGA-DX-A1L2, and TCGA-HB-A2OT[26,27].

Alignment and Variant Calling

Sequencing reads were aligned to the GRCh38.p7 reference build including decoy contigs from GATK using BWA-MEM2[29] (v2.2.1) in paired-end mode followed by duplicate marking with MarkDuplicatesSpark using GATK[30] (v4.2.4.1). For the GRCh37 runtime benchmarking, alignment was performed to the GRCh37 reference build including decoy contigs. The results alignments were recalibrated through Indel realignment using GATK (v3.7.0) and base-quality score recalibration using GATK (v4.2.4.1). Quality metrics were generated using SAMtools[19] (v1.18) stats and Picard[21] (v3.1.0) CollectWgsMetrics. Germline SNPs were called using HaplotypeCaller from GATK (v4.2.4.1) followed by variant recalibration using GATK (v4.2.4.1). Germline SVs were called using Delly2[31] (v1.2.6) and Manta[32] (v1.6.0). Mitochondrial SNVs were called using mitoCaller[3] (v1.0.0). Somatic SNVs were called using MuSE2[33] (v2.0.4), SomaticSniper[34] (v1.0.5.0), Strelka2[35] (v2.9.10), and Mutect2[30] (v4.5.0.0) followed by a consensus workflow to identify variants called by two or more callers using BCFtools[36] (v1.17) with quality control plots generated with BPG[37] (v7.1.0) and VennDiagram[38] (v1.7.4). Somatic SVs were called using Delly2[31] (v1.2.6) and Manta[32] (v1.6.0). Somatic CNAs were called using CNV_FACETS[39] (v0.16.0) for the PCAWG sample and using Battenberg[40] (v2.2.9) for the TCGA sample with visualization generated using BPG[37] (v7.1.0). Taking the consensus set of somatic SNV calls and the CNA calls, subclonal reconstruction was performed using PyClone-VI[41] (v0.1.2), PhyloWGS[42] (v2205be1), and FastClone[43] (v1.0.9). Reconstructed phylogeny was visualized using CEV[44] (v2.0.0). Data validation was performed with PipeVal[24] (v5.1.0) and data processing was done using Nextflow[13] (v23.04.2).

Acknowledgements

The authors gratefully acknowledge the ongoing support of all present and past members of the Boutros lab in providing suggestions, practical use-cases and support. The authors also acknowledge the Office of Health Informatics and Analytics at UCLA Health IT for their infrastructure support, particularly high-performance compute provisioning, data management and resource tuning.

Conflict of Interest Statement

PCB sits on the Scientific Advisory Boards of Intersect Diagnostics Inc., BioSymetrics Inc. and previously sat on that of Sage Bionetworks. All other authors have no conflicts of interest to declare.

Funding Sources

This study was conducted with the support of the National Institutes of Health through awards P30CA016042, R01CA244729, R01CA270108, U2CCA271894, U24CA248265 and U54HG012517, and of the Department of Defense through awards W81XWH2210247 and W81XWH2210751. NKW, HKW, JO, and CZ were supported by the Jonsson Comprehensive Cancer Center Fellowship. AEG was supported by the Howard Hughes Medical Institute Gilliam Fellowship. NZ was supported by the National Institute of Health through awards T32HG002536 and F31CA281168. LYL was supported by the Canadian Institutes of Health Research Vanier Fellowship and the Ontario Graduate Scholarship. SW was supported by the UCLA Tumor Cell Biology Training Program through the USHHS Ruth L. Kirschstein Institutional National Research Service Award T32CA009056. BN was supported by the National Library of Medicine T15LM013976 Training Grant and ASCO Young Investigator Award. BLT was supported by the UCLA Cancer Center Support Grant (P30CA016042) and the National Institutes of Health through awards U2CCA271894, U24CA248265, R01CA272678. RH was supported by EMBO Postdoctoral Fellowship ALTF 1131-2021 and the Prostate Cancer Foundation Young Investigator Award 22YOUN32.

References

1. Shendure, J., *et al*. (2017) DNA sequencing at 40: past, present and future. *Nature*, **550**, 345-353
2. Logsdon, Glennis A., *et al*. (2020) Long-read human genome sequencing and its applications. *Nature Reviews Genetics*, **21**, 597-614
3. Ding, J., *et al*. (2015) Assessing mitochondrial DNA variation and copy number in lymphocytes of ~2,000 Sardinians using tailored sequencing analysis tools. *PLOS Genetics*, **11**
4. Zhang, Y., *et al*. (2023) Association of Mitochondrial DNA Copy Number With Brain MRI Markers and Cognitive Function: A Meta-analysis of Community-Based Cohorts. *Neurology*, **100**, e1930-e1943
5. Gauthier, J., *et al*. (2019) A brief history of bioinformatics. *Briefings in Bioinformatics*, **20**, 1981-1996
6. Puttick, C., *et al*. (2024) MHC Hammer reveals genetic and non-genetic HLA disruption in cancer evolution. *Nature Genetics*, **56**, 2121-2131
7. Chakravarty, D., *et al*. (2021) Clinical cancer genomic profiling. *Nature Reviews Genetics*, **22**, 483-501
8. Sosinsky, A., *et al* (2024) Insights for precision oncology from the integration of genomic and clinical data of 13,880 tumors from the 100,000 Genomes Cancer Programme. *Nature Medicine*, **30**, 279-289
9. Cremin, C., *et al.* (2022) Big data: Historic advances and emerging trends in biomedical research. *Current Research in Biotechnology*, **4**, 138-151
10. The Galaxy Community. (2022) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Research*, **50**, W354-W351
11. Köster, J., *et al*. (2012) Snakemake – A scalable bioinformatics workflow engine. *Bioinformatics*, **28**, 2520-2522
12. Crusoe, M., *et al*. (2022) Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language. *Communications of the ACM*, **65**, 54-63
13. Di Tommaso, P., *et al.* (2017) Nextflow enables reproducible computational workflows. *Nature Biotechnology*, **35**, 316-319
14. Patel, Y., *et al*. (2024) NFTest: automated testing of Nextflow pipelines. *Bioinformatics*, **40**
15. Dash, S., *et al*. (2019) Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*, **6**, 54
16. Ellrott, K., *et al.* (2019) Reproducible biomedical benchmarking in the cloud: lessons from crowd-sourced data challenges. *Genome Biology*, **20**
17. Wilkinson, M. D., *et al*. (2016) The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, **3**, 160018
18. Cock, P., *et al*. (2009) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, **36**, 1767-1771
19. Li, H., *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078-2079

20. Salcedo, A., *et al*. (2024) Crowd-sourced benchmarking of single-sample tumor subclonal reconstruction. *Nature Biotechnology*
21. Broad Institute. (2019) Picard toolkit. *Broad Institute, GitHub repository*
22. Okonechnikov, K., *et al*. (2016) Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics*, **32**, 292-294
23. Sherry, S., *et al.* (1999) dbSNP – Database for Single Nucleotide Polymorphisms and Other Classed of Minor Genetic Variation. *Genome Res.*, **9**, 677-679
24. Patel, Y., *et al*. (2024) PipeVal: light-weight extensible tool for file validation. *Bioinformatics*, **40**
25. Yoo, A., *et al*. (2003) SLURM: Simple Linux Utility for Resource Management. *Lecture Notes in Computer Science*, **2862**
26. Abeshouse, A., *et al*. (2017) Comprehensive and Integrated Genomic Characterization of Adult Soft Tissue Sarcomas. *Cell*, **171**, 950-965
27. The ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium. (2020) Pan-cancer analysis of whole genomes. *Nature*, **578**, 82-93
28. Behera, S., *et al*. (2024) Comprehensive genome analysis and variant detection at scale using DRAGEN. *Nature Biotechnology*
29. Vasimuddin, M., *et al*. (2019) Efficient Architecture-Aware Acceleration of BWA-MEM for Multicore Systems. *IEEE Parallel and Distributed Processing Symposium*
30. McKenna A., *et al*. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res*, **20**, 1297-303
31. Rausch, T., *et al*. (2012) DELLY: structural variant discovery by integrated paired-end and split-end analysis. *Bioinformatics*, **28**, i333-i339
32. Chen, X., *et al.* (2016) Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, **32**, 1220-1222
33. Ji, S., *et al*. (2022) MuSE: A Novel Approach to Mutation Calling with Sample-Specific Error Modeling. *Methods Mol Biol*, **2493**, 21-27
34. Larson, D., *et al*. (2012) SomaticSniper: identification of somatic point mutations in whole genome sequencing data. *Bioinformatics*, **28**, 311-317
35. Kim, S., *et al*. (2018) Strelka2: fast and accurate calling of germline and somatic variants. *Nature Methods*, **15**, 591-594
36. Danecek, P., *et al*. (2021) Twelve years of SAMtools and BCFtools. *Gigascience*, **10**
37. P'ng, C., *et al*. (2019) BPG: Seamless, automated and interactive visualization of scientific data. *BMC Bioinformatics*, **20**
38. Chen, H., *et al*. (2011) VennDiagram: a package for the generation of highly-customizable Venn and Euler diagrams in R. *BMC Bioinformatics*, **12**
39. Shen, R., *et al*. (2016) FACETS: allele-specific copy number and clonal heterogeneity analysis tool for high-throughput DNA sequencing. *Nucleic Acids Research*, **44**
40. Nik-Zainal, S., *et al*. (2012) The life history of 21 breast cancers. *Cell*, **149**, 994-1007
41. Gillis, S., *et al*. (2020) PyClone-VI: scalable inference of clonal population structures using whole genome data. *BMC Bioinformatics*, **21**
42. Deshwar, A., *et al*. (2015) PhyloWGS: Reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. *Genome Biology*, **16**
43. Xiao, Y., *et al*. (2020) FastClone is a probabilistic tool for deconvoluting tumor heterogeneity in bulk-sequencing samples. *Nature Communications*, **11**
44. Winata, H., *et al*. (2024) CEV: Visualization of Cancer Evolution. *Manuscript in preparation*