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

Yash Patel[1,2,3,*], Chenghao Zhu[1,2,3,*], Takafumi N. Yamaguchi[1,2,3,*], Nicholas K. Wang[1,2,3], Nicholas Wiltsie[1,2,3], Alfredo E. Gonzalez[1,2,3], Helena K. Winata[1,2,3], Nicole Zeltser[1,2,3], Yu Pan[1,2,3], Mohammed Faizal Eeman Mootor[1,2,3], Timothy Sanders[1,2,3], Cyriac Kandoth[1,2,3], Sorel T. Fitz-Gibbon[1,2,3], Julie Livingstone[1,2,3], Lydia Y. Liu[1,2,3], Benjamin Carlin[1,2,3], Aaron Holmes[1,2,3], Jieun Oh[1,2,3], John Sahrmann[1,2,3], Shu Tao[1,2,3], Stefan Eng[1,2,3], Rupert Hugh-White[1,2,3], Kiarod Pashminehazar[1,2,3], Andrew Park[1,2,3], Arpi Beshlikyan[1,2,3], Madison Jordan[1,2,3], Selina Wu[1,2,3], Mao Tian[1,2,3], Jaron Arbet[1,2,3], Beth Neilsen[1,2,3], Yuan Zhe Bugh[1,2,3], Gina Kim[1,2,3], Joseph Salmingo[1,2,3], Wenshu Zhang[1,2,3], Roni Haas[1,2,3], Aakarsh Anand[1,2,3], Edward Hwang[1,2,3], Anna Neiman-Golden[1,2,3], Philippa Steinberg[1,2,3], Wenyan Zhao[1,2,3], Prateek Anand[1,2,3], Brandon L. Tsai[1,2,3], 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 is becoming more affordable and faster through advances in high-throughput technologies. This rise in data availability has contributed to the development of novel algorithms to elucidate previously obscure features and led to an increased reliance on complex workflows to integrate such tools into analyses pipelines. To facilitate the analysis of DNA sequencing data, we created metapipeline-DNA, a highly configurable and extensible pipeline. It encompasses a broad range of processing including raw sequencing read alignment and recalibration, variant calling, quality control and subclonal reconstruction. Metapipeline-DNA also contains configuration options to select and tune analyses while being robust to failures. This standardizes and simplifies the ability to analyze large DNA sequencing in both clinical and research settings.

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

Introduction

With the rapid progression in efficiency and affordability of high-throughput technologies, biomedical research has seen a sharp increase in the generation and volume of large sequencing datasets. As DNA sequencing becomes more cost-efficient and rapid, it is increasingly used in both routine clinical care and for research studies (Shendure, *et al*., 2017). Technical advances have also facilitated studying of previously obscure features. The development of long-read sequencing through nanopores, for example, has given insights into structural variants (SVs) and complex repetitive regions of the genome that were difficult to capture through traditional short-read sequencing (Branton, *et al*., 2008). Given such possibilities, the number of features being studied through sequencing has greatly expanded, with research on any given dataset studying variant calling of single-nucleotide variants (SNVs) and SVs, telomere length and dynamics, mitochondrial genome sequencing and calling, mutational signatures and so forth (Ding, *et al*., 2015; Gauthier, *et al.*, 2019).

The availability of such data has been paralleled by development and use of software for processing and analysis in both research and clinical settings, with new discoveries relying heavily on complex workflows comprising established and novel algorithms (Cremin, *et al*., 2022). These workflows, often referred to as “pipelines”, are implemented through a range of orchestration frameworks built for data processing to minimize manual handling of data flow and facilitate the stitching together of different tools and algorithms to ultimately process raw data into more refined forms. Widely-used orchestration frameworks in computational biology include Galaxy, Snakemake, Common Workflow Language (CWL), and Nextflow (Crusoe, *et al*., 2022; Köster, *et al*., 2012; The Galaxy Community, 2022; Di Tommaso, *et al.*, 2017).

The use of complex workflows has placed a growing emphasis on standardization, extensibility, quality control, and compute infrastructure needs. Workflow implementations are routinely different from team to team, and often lack critical features like configuration with multiple algorithms to facilitate use of specialized tools, automated quality control and visualization to maintain integrity and quality of data, testability, and automated recovery from failure (Patel, *et al*., 2024a). Given the volume of data and the necessary compute, workflows are often designed for high-performance computing environments which may vary across different providers (Marx, 2013). This brings up a need for cross-provider compatibility and portability of workflows for new environments, a concept aligning with the “model to data” (M2D) paradigm in data sharing and processing (Ellrott, *et al.*, 2019). Rather than shuffling data, which is infeasible due to data size and privacy, around across infrastructure for processing, M2D instead relies on bringing the model or algorithms to the data, thus necessitating that models be portable across providers and environments.

To address this need for a robust sequencing analysis pipeline, we created metapipeline-DNA, a highly configurable DNA sequencing analysis pipeline capable of processing data from any stage of analysis up to and including subclonal reconstruction. It encompasses steps to process DNA sequencing data starting with raw reads, perform alignment and recalibration, call variants, and perform subclonal reconstruction with quality control built into the workflow level and the individual steps. It includes a broad range of configuration options for selecting and tuning analyses including support for robustly picking up analysis from failed runs without having to restart the entire workflow.

Results

Overview

Metapipeline-DNA is a Nextflow meta-pipeline for analysis of DNA sequencing starting from raw sequencing reads and including all major classes of variant detection (**Figure 1A**). It encompasses 12 pipelines (**Table 1**), each of which can be executed independently. All pipelines are extensively parameterized through configuration which allows for customization, selection, and tuning of algorithms with available options. Individual pipelines can allow execution of multiple algorithms and even create consensus calls from them. For example, four separate algorithms can be executed for somatic single nucleotide variant (SNV) detection, generating a consensus set of predictions and associated data-visualizations (**Figure 1B;** Patel, *et al*., 2024a).

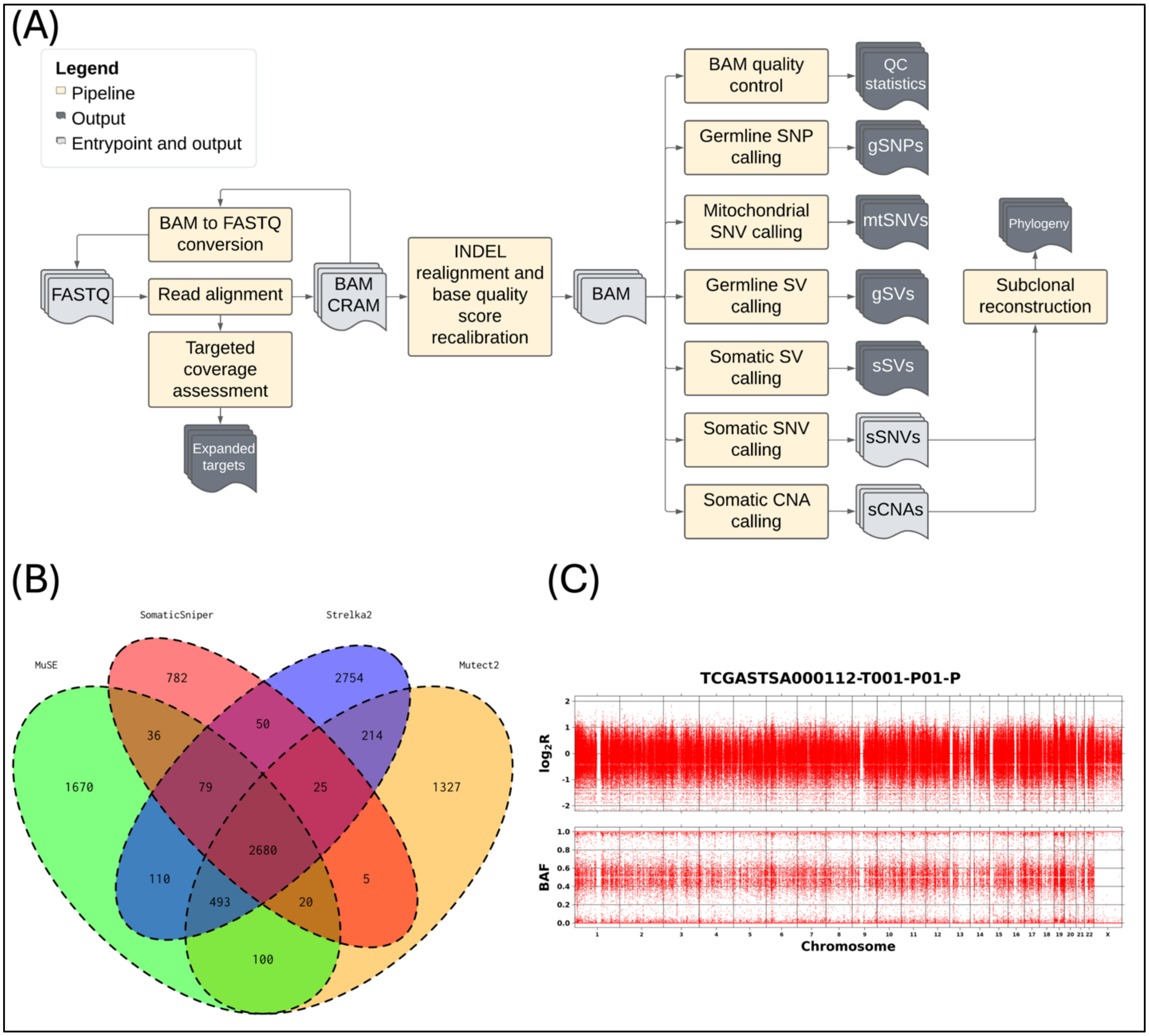


Figure 1. Data flow and visualizations. A. Data flow through metapipeline-DNA. B. Example intersection diagram of consensus variants between 4 SNV callers: MuSE2, SomaticSniper, Strelka2, and Mutect2. C. Normalized tumour coverage relative to the matched normal ($\log_2 R$) and the B-allele frequency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

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	SAMtools BEDtools	Automatic expansion of regions to off-target dnSNP loci with coverage

		metrics		
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 calls* *Panel of normal*	Somatic SNV VCFs	Mutect2 Strelka2 SomaticSniper MuSE BCFtools-Intersect	Support for panel of normals Tumour-only mode Multi-tumour mode Consensus callset and vizualization
Call-sCNA	BAM	Somatic CNA VCF or TSV	Battenberg FACETS	Standardized visualization of aberrations 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.

Pipeline	**PCAWG (runtime in hours)**	**PCAWG (Peak RAM in GB)**	**TCGA (runtime in hours)**	**TCGA (Peak RAM in GB)**
Align-DNA (normal)	4.62	55.3	0.27	24.3
Align-DNA (tumour)	8.93	55.9	0.32	24.7
Recalibrate-BAM	30.40	20.6	1.63	2.8
Generate-SQC-BAM	6.23	1.5	0.28	0.96
Call-gSNP	8.90	5.4	0.37	5.1
Call-mtSNV	4.15	8.1	0.13	6.4
Call-sSNV	13.12	42	0.37	31.5
Call-sSV	17.88	12.9	0.37	8
Call-gSV	8.98	6	0.23	2
Call-sCNA	4.15	45	2.68	19.1
Call-SRC	1.40	0.37	0.03	0.25
TOTAL	**72.2**	**-**	**5.35**	**-**

Table 2: Runtime of pipelines per sample. The total runtime is less than the sum of the individual pipelines' runtimes due to parallelization of variant calling pipelines.

The standard run-time mode accepts input sequencing data in FASTQ format and executes all pipelines on it starting with alignment. Aligned and unaligned BAM and CRAM files can also be used as entry-points, with automated BAM-to-FASTQ conversions performed as needed (Cock, *et al*., 2009; Li, *et al*., 2009). A few pipelines can accept alternative entry-points, such as direct use of SNV and copy number aberration (CNA) calls for tumour subclonal reconstruction**.** All dependencies, input, and output formats are available on well-structured and standardized GitHub pages for the respective pipeline.

We designed metapipeline-DNA to be intrinsically flexible. Users can select any subset of analyses for execution, and all necessary dependencies are automatically identified and executed. All run-modes and dependency identification have defaults set to the most common behaviour, but with parameters available for easy configuration. For example, when the input data is aligned, options exist to control whether reads are back-converted to FASTQ and then re-aligned, whether the aligned reads undergo recalibration, or whether an input BAM or CRAM is directly used for downstream analyses.

Several different sample run-modes are available, which we denote using the terminology nT-mN, where T indicates the number of tumour samples and N the number of reference samples. Thus the classic paired tumour-normal analysis mode is 1T-1N. Metapipeline-DNA fully supports modes like 0T-1N (*i.e.* germline DNA sequencing), 0T-3N (*e.g.* familial trios), 1T-0N (*i.e.* unpaired tumour-only sequencing) and arbitrary multi-region tumour and/or reference sequencing. The primary limitation to multi-sample analyses are 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 only germline structural variant detection is attempted, independent of user-selections.

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 WGS mode and targeted-sequencing mode, based on user parameterization. Targeted-sequencing model supports all subsets of the whole genome, including whole-exome sequencing. Options are available to assess coverage, expand targets with off-target coverage sites, and automatically use expanded target intervals for downstream processing.

Data Visualization & Quality-Control

Metapipeline-DNA includes a range quality control steps and pipelines to assess data quality. BAM quality is assessed with alignment and coverage metrics. In targeted-sequencing mode, coverage assessment is performed through per-base depth calculations at target regions and well-characterized off-target polymorphic sires from dbSNP. Pipelines also perform specific quality control for cross-individual contamination and variant-type specific metrics (**Figure 1B, 1C**).

Software-Engineering & Pipeline Robustness

Our pipeline development placed a heavy focus on generating re-usable and extensible software that could automatically detect and recover from common errors. 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 (Patel, *et al*., 2024a). Metapipeline-DNA has a suite of 71 total 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. The development effort to-date has involved 42 contributors making 1220 pull-requests, and 45 individuals making 973 suggestions, feature-requests and issue-reports.

Bioinformatics data has high intrinsic variability, and bioinformatics software can be prone to significant numbers of failures – particularly in heterogeneous 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 (Patel, *et al*., 2024b). 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. 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. 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 cloud-agnostic workflow, with successful testing and validation performed in both 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 performed with the Slurm executor with option available to select the specific compute partitions used to run analyses (Yoo, *et al*., 2003). 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 requiring intervention. Real-time monitoring is also 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 with eager intermediate file removal and built in checks to allow for optimized disk usage (performing I/O operations from high-performance working disks) without losing any output data. Resource allocation for individual steps is also automatically handled, with steps from pipelines running in parallel filling in available resources as available. Resource-related robustness is also built into pipelines to detect shortages in memory allocation and automatically retry processes with higher allocations.

Use-Case: PCAWG-63 Breast cancer normal-tumour pair and TCGA sarcoma normal-tumour pair

As a demonstration, two normal-tumour pairs were processed through the entirety of metapipeline-DNA. One pair was selected from the Pan-cancer Analysis of Whole Genomes (PCAWG) 63 dataset and the other from The Cancer Genome Atlas

(Abeshouse, *et al*., 2017; The ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium, 2020). The PCAWG-63 sample was derived from a breast cancer sample sequenced with whole-genome sequencing. The TCGA samples was derived from a soft tissue sarcoma sample sequenced with exome-targeted sequencing. Both pairs were processed using metapipeline-DNA from alignment to subclonal reconstruction. Runtimes of metapipeline-DNA for these samples are summarized in **Table 2**.

Discussion

Metapipeline-DNA is designed to facilitate the analysis of DNA sequencing data in a highly configurable and robust manner with support for a broad range of variant calling and analyses. The volume of available sequencing data is rapidly increasing, bringing with it development in tools and algorithms designed to study features from the raw data. Metapipeline-DNA collects and encompasses a range of algorithms to ease the multi-step analyses often carried out with sequencing data. The design also allows for a high level of customizability to select different algorithms for different processing.

The high level of customizability brings with it the ability to expand the set of algorithms and pipelines available within metapipeline-DNA. Individual pipelines within the meta-pipeline 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. Specialized algorithms designed to leverage hardware, such as FPGAs and GPUs, outside of the standard CPUs and RAM do exist for processing sequencing data. The integration of such tools along with support for executing processes on specialized hardware remain to be incorporated into metapipeline-DNA and will be made possible by the modular nature of pipelines. The context of DNA also brings up the possibility of similar meta-pipelines for other biological molecules such as RNA and proteins. 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 being generated and processed in sequencing studies is often very large. With that comes a need for optimization of analyses 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. There are additional enhancements that are underway to minimize duplicated data and disk usage of metapipeline-DNA.

Methods

Analysis Cohort

To demonstrate the use of metapipeline-DNA, we chose two normal-tumour pairs: one WGS breast cancer pair from PCAWG-63 donor DO2629 and one exome sequencing soft tissue sarcoma pair from TCGA donor TCGA-QQ-A8VD (Abeshouse, *et al*., 2017; The ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium, 2020).

Alignment and variant calling

Sequencing reads were aligned to the GRCh38 reference build including decoy contigs from GATK using BWA-MEM2 (v2.2.1) in paired-end mode followed by duplicate marking with MarkDuplicatesSpark using GATK (v4.2.4.1) (McKenna, *et al*., 2010; Vasimuddin, *et al*., 2019). 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 (v1.18) stats and Picard (v3.1.0) CollectWgsMetrics (Broad Institute, 2019; Li, *et al*., 2009). 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 (v1.2.6) and Manta (v1.6.0) (Chen, *et al*., 2016; Rausch, *et al*., 2012). Mitochondrial SNVs were called using mitoCaller (v1.0.0) (Ding, *et al*., 2015). Somatic SNVs were called using MuSE2 (v2.0.4), SomaticSniper (v1.0.5.0), Strelka2 (v2.9.10), and Mutect2 (v4.5.0.0) followed by a consensus workflow to identify variants called by 2 or more callers using BCFtools (v1.17) (Danecek, *et al*., 2021; Ji, *et al*., 2022; Kim, *et al*., 2018; Larson, *et al*., 2012). Somatic SVs were called using Delly2 (v1.2.6) and Manta (v1.6.0). Somatic CNAs were called using CNV_FACETS (v0.16.0) for the PCAWG sample and using Battenberg (v2.2.9) for the TCGA sample (Nik-Zainal, *et al*., 2012; Shen, *et al.*, 2016). Taking the consensus set of somatic SNV calls and the CNA calls, subclonal reconstruction was performed using PyClone-VI (v0.1.2), PhyloWGS (v2205be1), and FastClone (v1.0.9) (Deshwar, *et al*., 2015; Gillis, *et al*., 2020; Xiao, *et al*., 2020). Data validation was performed with PipeVal (v5.1.0) and data processing was done using Nextflow (v23.04.2) (Patel, *et al*., 2024b).

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, and JO 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.

References

Abeshouse, A., *et al*. (2017) Comprehensive and Integrated Genomic Characterization of Adult Soft Tissue Sarcomas. *Cell*, **171**, 950-965

Branton, D., *et al*. (2008) The potential and challenged of nanopore sequencing. *Nature Biotechnology*, **10**, 1146-53

Broad Institute. (2019) Picard toolkit. *Broad Institute, GitHub repository*

Chen, X., *et al.* (2016) Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, **32**, 1220-1222

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

Cremin, C., *et al.* (2022) Big data: Historic advances and emerging trends in biomedical research. *Current Research in Biotechnology*, **4**, 138-151

Crusoe, M., *et al*. (2022) Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language. *Communications of the ACM*, **65**, 54-63

Danecek, P., *et al*. (2021) Twelve years of SAMtools and BCFtools. *Gigascience*, **10**

Deshwar, A., *et al*. (2015) PhyloWGS: Reconstructing subclonal composition and evolution from whole-genome sequencing of tumors. *Genome Biology*, **16**

Di Tommaso, P., *et al.* (2017) Nextflow enables reproducible computational workflows. *Nature Biotechnology*, **35**, 316-319

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**

Ellrott, K., *et al.* (2019) Reproducible biomedical benchmarking in the cloud: lessons from crowd-sourced data challenges. *Genome Biology*, **20**

Gauthier, J., *et al*. (2019) A brief history of bioinformatics. *Briefings in Bioinformatics*, **20**, 1981-1996

Gillis, S., *et al*. (2020) PyClone-VI: scalable inference of clonal population structures using whole genome data. *BMC Bioinformatics*, **21**

Ji, S., *et al*. (2022) MuSE: A Novel Approach to Mutation Calling with Sample-Specific Error Modeling. *Methods Mol Biol*, **2493**, 21-27

Kim, S., *et al*. (2018) Strelka2: fast and accurate calling of germline and somatic variants. *Nature Methods*, **15**, 591-594

Köster, J., *et al.* (2012) Snakemake – A scalable bioinformatics workflow engine. *Bioinformatics*, **28**, 2520-2522

Larson, D., *et al*. (2012) SomaticSniper: identification of somatic point mutations in whole genome sequencing data. *Bioinformatics*, **28**, 311-317

Li, H., *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078-2079

Marx, V. (2013) The big challenges of big data. *Nature*, **498**, 255-260

McKenna A., *et al*. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res*, **20**, 1297-303

Nik-Zainal, S., *et al*. (2012) The life history of 21 breast cancers. *Cell*, **149**, 994-1007

Patel, Y., *et al*. (2024) NFTest: automated testing of Nextflow pipelines. *Bioinformatics*, **40**

Patel, Y., *et al*. (2024) PipeVal: light-weight extensible tool for file validation. *Bioinformatics*, **40**

Rausch, T., *et al*. (2012) DELLY: structural variant discovery by integrated paired-end and split-end analysis. *Bioinformatics*, **28**, i333-i339

Shen, R., *et al*. (2016) FACETS: allele-specific copy number and clonal heterogeneity analysis tool for high-throughput DNA sequencing. *Nucleic Acids Research*, **44**

Shendure, J., *et al*. (2017) DNA sequencing at 40: past, present and future. *Nature*, **550**, 345-353

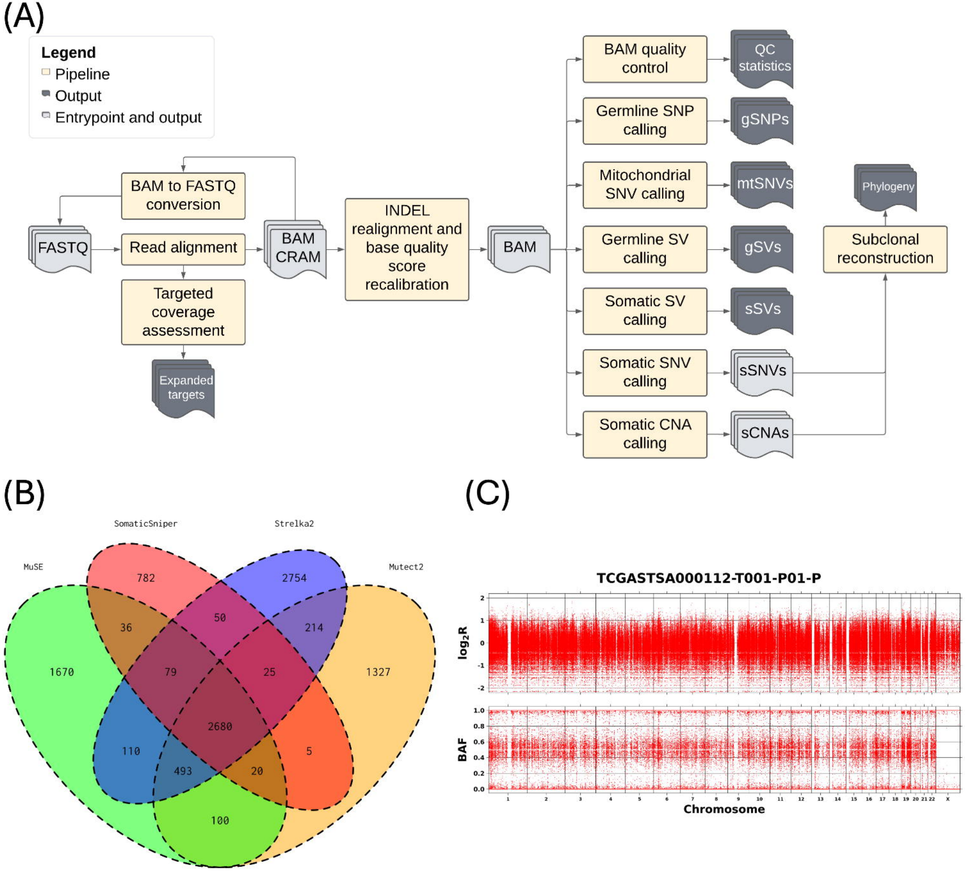
The Galaxy Community. (2022) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Research*, **50**, W354-W351

The ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium. (2020) Pan-cancer analysis of whole genomes. *Nature*, **578**, 82-93

Vasimuddin, M., *et al*. (2019) Efficient Architecture-Aware Acceleration of BWA-MEM for Multicore Systems. *IEEE Parallel and Distributed Processing Symposium*

Xiao, Y., *et al*. (2020) FastClone is a probabilistic tool for deconvoluting tumor heterogeneity in bulk-sequencing samples. *Nature Communications*, **11**

Yoo, A., *et al*. (2003) SLURM: Simple Linux Utility for Resource Management. *Lecture Notes in Computer Science*, **2862**


(A)
Legend
Pipeline
Output
Entrypoint and output
FASTQ
BAM to FASTQ conversion
Read alignment
Targeted coverage assessment
Expanded targets
BAM CRAM
INDEL realignment and base quality score recalibration
BAM
BAM quality control
QC statistics
Germline SNP calling
gSNPs
Mitochondrial SNV calling
mtSNVs
Germline SV calling
gSVs
Somatic SV calling
sSVs
Somatic SNV calling
sSNVs
Somatic CNA calling
sCNAs
Subclonal reconstruction
Phylogeny
(B)
SomaticSniper
Strelka2
MuSE
Mutect2
782
2754
36
50
214
1670
79
25
1327
2680
110
5
493
20
100
(C)
TCGASTSA000112-T001-P01-P
log2R
2
1
0
-1
-2
BAF
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
X
Chromosome