Bioinformatics, 2022, 1–3
https://doi.org/10.1093/bioinformatics/btac600
Advance Access Publication Date: 2 September 2022
Applications Note

Genome analysis

grenepipe: a flexible, scalable and reproducible pipeline to automate variant calling from sequence reads

Lucas Czech [1,*] **and Moises Exposito-Alonso** [1,2,3,*]

[1]Department of Plant Biology, Carnegie Institution for Science, Stanford, CA 94305, USA, [2]Department of Global Ecology, Carnegie Institution for Science, Stanford, CA 94305, USA and [3]Department of Biology, Stanford University, Stanford, CA 94305, USA

*To whom correspondence should be addressed.

Associate Editor: Christina Kendziorski

Received on March 31, 2022; revised on July 27, 2022; editorial decision on August 14, 2022; accepted on September 5, 2022

Abstract

Summary: We developed grenepipe, an all-in-one Snakemake workflow to streamline the data processing from raw high-throughput sequencing data of individuals or populations to genotype variant calls. Our pipeline offers a range of popular software tools within a single configuration file, automatically installs software dependencies, is highly optimized for scalability in cluster environments and runs with a single command.

Availability and implementation: grenepipe is published under the GPLv3 and freely available at github.com/moiexpositoalonsolab/grenepipe.

Contact: luc.czech@gmail.com or moisesexpositoalonso@gmail.com

1 Introduction

High-throughput sequencing technologies have revolutionized biomedical, ecological and evolutionary research. Whether sequencing is conducted on single cells, or on pooled DNA of whole populations, the core bioinformatic processing is virtually the same: the sequencing reads (typically 30–250 letters long) are compared to a reference sequence or genome to find the variant positions, e.g. single nucleotide polymorphisms (SNPs) or insertions/deletions (indels). This process requires bioinformatic expertise to manipulate large datasets and manage a plethora of software dependencies. Because this is often a limitation for novices, we aimed to streamline the processing with grenepipe, an automated and flexible pipeline for variant (and frequency) calling from raw sequences. Its backend is the platform-independent Snakemake workflow engine (Köster and Rahmann, 2012; Mölder *et al.*, 2021). This allows it to process large datasets, takes care of intermediate file bookkeeping and execution order dependencies and parallelizes independent jobs. All software tools are automatically installed via Conda/Bioconda (Grüning *et al.*, 2018), the execution can seamlessly recover and continue after failed jobs, and workflows and results can be archived to facilitate reproducibility.

Although several workflows for such analyses exist (Chiang *et al.*, 2015; Cokelaer *et al.*, 2017; Lataretu and Hölzer, 2020; Singer *et al.*, 2018; https://github.com/snakemake-workflows/dna-seq-gatk-variant-calling), our pipeline focuses on automation and simplicity, and it can run with a single one-line command. It comprises all steps of the well-established GATK best-practices workflow (DePristo *et al.*, 2011) and adds recent popular tools and further quality controls with a focus on evolutionary ecology applications, including quality profiling of ancient DNA from historical specimens (Fellows Yates *et al.*, 2021), and within-library calculation of allele frequencies for Pool-Sequencing (Schlötterer *et al.*, 2014). For most steps, we offer to choose between several distinct tools, with several additional optional steps. Each tool can be freely configured as needed, allowing the pipeline to be used as an exploratory tool to assess outcomes under different parameterizations, or towards obtaining consensus sets of variants. A high-level overview of the pipeline is shown in Figure 1.

2 Pipeline overview and steps

2.1 Input

The input to the pipeline is FASTQ files (Cock *et al.*, 2010) representing distinct samples, either single-end or paired-end, potentially gzip-compressed. Samples can also consist of multiple units of FASTQ files representing, e.g. re-sequencing runs of the same biological sample. In addition, a reference genome or sequence to compare with is required in FASTA format (Pearson and Lipman, 1988). Optionally, if the species genome is annotated in the SnpEff (Cingolani *et al.*, 2012) or VEP (McLaren *et al.*, 2016) databases, a suitable species name can be provided to run these tools. Lastly, if the user has a known set of genomic positions of interest, a reference VCF can be provided for guided SNP calling.

2.2 Reference preparation

Initially, the pipeline needs to prepare the reference genome to enable parallelization of the workload across chromosomes or scaffolds of the reference genome. In this step, we run BWA INDEX (Li and Durbin, 2009; Li *et al.*, 2009), SAMTOOLS FAIDX (Li *et al.*, 2009), GATK CreateSequenceDictionary (Genome Analysis Toolkit) (McKenna *et al.*, 2010) and Tabix (Li, 2011b), to create indices of

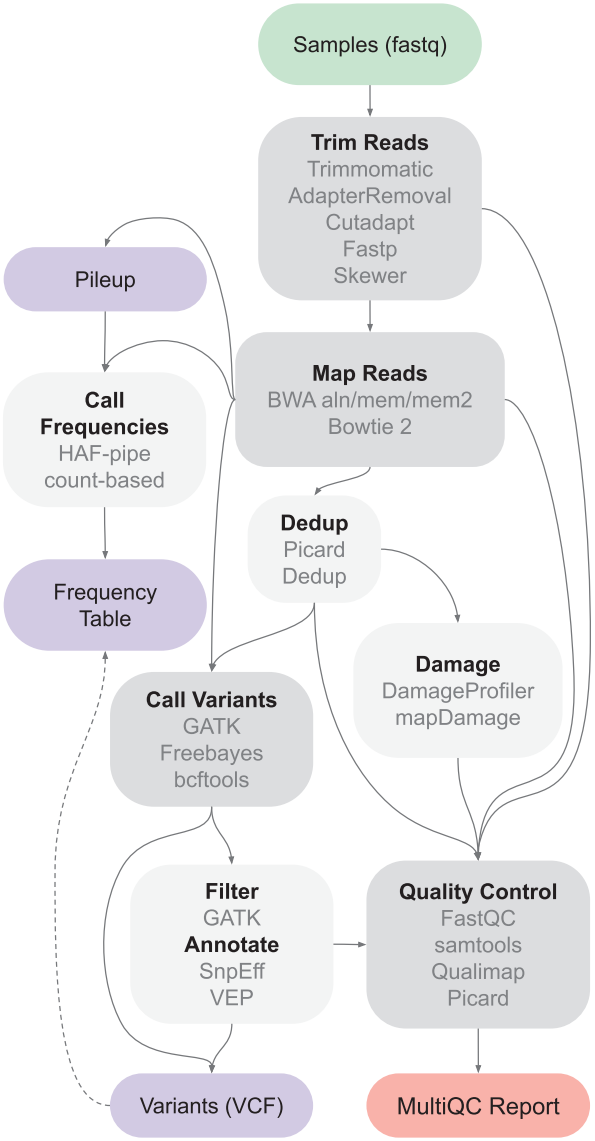

Fig. 1. grenepipe workflow: This high-level overview exemplifies the data flow of grenepipe, from raw sequencing data (amples FASTQ) to quality control report (MultiQC report) and final variant (or frequency) output (variants VCF, frequency table). Main steps are in dark gray, and optional data-type-specific steps in light gray. For further details, visit the online wiki at: github.com/moiexpositoalonsolab/grenepipe (A color version of this figure appears in the online version of this article.)

the reference genome, as well as SeqKit (Shen *et al.*, 2016) to provide statistics of the reference genome.

2.3 Read trimming

Read trimming removes adapter sequences and low-quality bases. Typical read trimmers can operate in either single-end or paired-end mode. Some tools can furthermore merge paired-end reads into a single read. Users can select from several read trimming tools: AdapterRemoval (Lindgreen, 2012; Schubert *et al.*, 2016), Cutadapt (Martin, 2011), fastp (Chen *et al.*, 2018), SeqPrep (https://github.com/jstjohn/SeqPrep), Skewer (Jiang *et al.*, 2014) and trimmomatic (Bolger *et al.*, 2014). The input and the output of this step are `fastq.gz` files.

2.4 Read mapping

Next, the reads are aligned/mapped against the reference genome. The pipeline uses bwa mem, bwa aln (Li and Durbin, 2009), bwa mem2 (Vasimuddin *et al.*, 2019) or Bowtie2 (Langmead and Salzberg, 2012) for this step, creating SAM/BAM files. Duplicated reads can occur in the data as an artifact of library preparation. We include the tools Picard MarkDuplicates (http://broadinstitute.github.io/picard/) and dedup (Peltzer *et al.*, 2016) to tag duplicate reads. The pipeline optionally allows for filtering the mapped reads, clipping overlaps between read pairs with BamUtil (Jun *et al.*, 2015), and re-calibrating base quality scores using GATK BaseRecalibrator (McKenna *et al.*, 2010), in order to detect systematic base call inaccuracies. Lastly, `(m)pileup` files can be generated for external processing.

2.5 Ancient DNA and damage profiling

DNA degradation and fragmentation are important considerations for libraries produced from historical or ancient specimens. We include two optional tools for estimating damage patterns, mapDamage (Ginolhac *et al.*, 2011; Jónsson *et al.*, 2013) and DamageProfiler (Neukamm *et al.*, 2020).

2.6 Quality control

Quality control assesses the sequence data to find issues in the library preparation and sequencing protocol. We include several dedicated tools, namely FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc), samtools stats and samtools flagstat (Li *et al.*, 2009), and Qualimap (Okonechnikov *et al.*, 2016), as well as Picard CollectMultipleMetrics. Furthermore, many tools described above report statistics about their output. All these results are compiled into a report by MultiQC (Ewels *et al.*, 2016), allowing researchers to assess quality control statistics and to examine individual samples as needed.

2.7 Variant calling

The core step of the pipeline is to identify ('call') genomic positions where one or more samples differ from the reference sequence; see (Olson *et al.*, 2015; Xu, 2018) for reviews and best practices. In this step, we offer GATK HaplotypeCaller (McKenna *et al.*, 2010), FreeBayes (Garrison and Marth, 2012; Neph *et al.*, 2012) or bcftools call (Li, 2011a). Note that ploidy can be specified when using FreeBayes or bcftools when studying species that are not diploid. Optionally, an input VCF file can be provided to restrict the calling to known variants (Neph *et al.*, 2012). Other tools, e.g. tools specialized for somatic variant calling, might be added in the future.

Subsequently, we filter SNPs and indels with GATK SelectVariants, while allowing for additional filtering using GATK VariantFiltration, or using GATK VariantRecalibrator (variant quality score recalibration). The outcome of these steps is the main VCF file (Danecek *et al.*, 2011) containing the variants for all samples, which can then optionally be annotated with SnpEff (Cingolani *et al.*, 2012) and VEP (McLaren *et al.*, 2016) to predict the effects of variants on genes.

2.8 Frequency calling

Although grenepipe is agnostic to the genomic application, an important use is Pool-Seq for eco-evolutionary studies, where DNA of a population is combined ('pooled') in the same sequencing library. Allele frequencies, rather than genotype states, can be extracted from the VCF file or directly from BAM files using our complementary tool grenedalf (https://github.com/lczech/grenedalf); this lists frequencies of biallelic SNPs of each library based on base ratios within samples for downstream computations. Furthermore, HAF-pipe (Kessner *et al.*, 2013; Tilk *et al.*, 2019) is integrated into the pipeline, which computes allele frequencies of pool-sequenced samples based on haplotype frequencies of the founder generation in Evolve-and-Resequence experiments.

2.9 Output

The main output is the variant call table (VCF file), as well as the MultiQC quality control report. Furthermore, relevant intermediate files (e.g. trimmed FASTQ files and mapped SAM/BAM files) are kept by default for further inspection or use in downstream analyses. Lastly, Snakemake can generate benchmarking reports, allowing users to evaluate tool runtimes and resource requirements.

3 Automation and distribution of jobs

All independent steps (e.g. individual samples) are run in parallel, maximizing throughput, e.g. on computer clusters. We further not only split data into chromosomes (or scaffolds) to improve the throughput of the variant calling but also implemented a feature that groups small contigs for improving computational efficiency.

A major advantage of grenepipe's Snakemake backend is its ability to recover from failed cluster jobs, e.g. network issues or broken hardware in cluster environments. By default, grenepipe continues with independent jobs as long as possible, and after fixing any issues, is able to proceed seamlessly from the jobs that already succeeded.

4 Example one-liner

Here and in the online wiki, we showcase the use of grenepipe with a toy dataset from pooled plant populations (Czech *et al.*, 2022). After preparing the samples table and configuration file, the following single command suffices to produce the final `VCF`:

```
# Produce vcf and MultiQC from example/samples/*.fastq
snakemake --use-conda --cores 4 --directory example/
```

The configuration file is where the tools to be used are selected, and their parameters are described and set.

Acknowledgements

We like to thank Moi Lab members, and Patricia Lang, Callie Rodgers Chappell, Nicolas Alexandre, Jay Yeam, Peter Pellitier, Shannon Hateley and Yunru Peng for their input on the project, and for their persistence as beta-testers. The soundtrack for this work was provided by Koji Kondo. This project was initiated as part of a collaborative network, Genomics of rapid Evolution to Novel Environments (GrENE-net.org), from which it inherits its name.

Funding

The Carnegie Institution for Science.

Data availability

grenepipe is published under the GPLv3 and is freely available at github.com/moiexpositoalonsolab/grenepipe/.

References

Bolger, A.M. *et al.* (2014) Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, **30**, 2114–2120.

Chen, S. *et al.* (2018) Fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics*, **34**, i884–i890.

Chiang, C. *et al.* (2015) SpeedSeq: ultra-fast personal genome analysis and interpretation. *Nat. Methods.*, **12**, 966–968.

Cingolani, P. *et al.* (2012) A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of drosophila melanogaster strain w1118; iso-2; iso-3. *Fly. (Austin)*, **6**, 80–92.

Cock, P.J.A. *et al.* (2010) The sanger FASTQ file format for sequences with quality scores, and the solexa/illumina FASTQ variants. *Nucleic Acids Res.*, **38**, 1767–1771.

Cokelaer, T. *et al.* (2017) 'Sequana': a set of snakemake NGS pipelines. *J. Open Source Softw.*, **2**, 352.

Czech, L. *et al.*; GrENE-net Consortium. (2022). *Monitoring rapid evolution of plant populations at scale with pool-sequencing. bioRxiv.* Cold Spring Harbor Laboratory, p. 2022.02.02.477408.

Danecek, P. *et al.*; 1000 Genomes Project Analysis Group. (2011) The variant call format and VCFtools. *Bioinformatics*, **27**, 2156–2158.

DePristo, M.A. *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.*, **43**, 491–498.

Ewels, P. *et al.* (2016) MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, **32**, 3047–3048.

Fellows Yates, J.A. *et al.* (2021) Reproducible, portable, and efficient ancient genome reconstruction with nf-core/eager. *PeerJ*, **9**, e10947.

Garrison, E. and Marth, G. (2012) *Haplotype-based variant detection from short-read sequencing*. arXiv. Cornell University.

Ginolhac, A. *et al.* (2011) mapDamage: testing for damage patterns in ancient DNA sequences. *Bioinformatics*, **27**, 2153–2155.

Grüning, B. *et al.*; The Bioconda Team. (2018) Bioconda: a sustainable and comprehensive software distribution for the life sciences. *Nat. Methods.*, **15**, 475–476.

Jiang, H. *et al.* (2014) Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC Bioinformatics*, **15**, 182.

Jónsson, H. *et al.* (2013). mapDamage2.0 Fast approximate Bayesian estimates of ancient DNA damage parameters. *Bioinformatics*, **29**, 1682–1684.

Jun, G. *et al.* (2015) An efficient and scalable analysis framework for variant extraction and refinement from population-scale DNA sequence data. *Genome Res.*, **25**, 918–925.

Kessner, D. *et al.* (2013) Maximum likelihood estimation of frequencies of known haplotypes from pooled sequence data. *Mol. Biol. Evol.*, **30**, 1145–1158.

Köster, J. and Rahmann, S. (2012) Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics*, **28**, 2520–2522.

Langmead, B. and Salzberg, S.L. (2012) Fast gapped-read alignment with bowtie 2. *Nat. Methods.*, **9**, 357–359.

Lataretu, M. and Hölzer, M. (2020) RNAflow: an effective and simple RNA-Seq differential gene expression pipeline using nextflow. *Genes*, **11**, 1487. https://www.mdpi.com/2073-4425/11/12/1487.

Li, H. (2011a) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, **27**, 2987–2993.

Li, H. (2011b) Tabix: fast retrieval of sequence features from generic TAB-delimited files. *Bioinformatics*, **27**, 718–719.

Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

Li, H. *et al.*; 1000 Genome Project Data Processing Subgroup (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Lindgreen, S. (2012) AdapterRemoval: easy cleaning of next-generation sequencing reads. *BMC Res. Notes.*, **5**, 337.

Martin, M. (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. J.*, **17**, 10.

McKenna, A. *et al.* (2010) The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, **20**, 1297–1303.

McLaren, W. *et al.* (2016) The ensembl variant effect predictor. *Genome Biol.*, **17**, 122.

Mölder, F. *et al.* (2021) Sustainable data analysis with snakemake. *F1000Research*, **10**, 33.

Neph, S. *et al.* (2012) BEDOPS: high-performance genomic feature operations. *Bioinformatics*, **28**, 1919–1920.

Neukamm, J. *et al.* (2020) *DamageProfiler: fast damage pattern calculation for ancient DNA*. bioRxiv. Cold Spring Harbor Laboratory, pp. 1–10.

Okonechnikov, K. *et al.* (2016) Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics*, **32**, 292–294.

Olson, N.D. *et al.* (2015) Best practices for evaluating single nucleotide variant calling methods for microbial genomics. *Front. Genet.*, **6**, 235.

Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, **85**, 2444–2448.

Peltzer, A. *et al.* (2016) EAGER: efficient ancient genome reconstruction. *Genome Biol.*, **17**, 60.

Schlötterer, C. *et al.* (2014) Sequencing pools of individuals – mining genome-wide polymorphism data without big funding. *Nat. Rev. Genet.*, **15**, 749–763.

Schubert, M. *et al.* (2016) AdapterRemoval v2: rapid adapter trimming, identification, and read merging. *BMC Res. Notes*, **9**, 88.

Shen, W. *et al.* (2016) SeqKit: a cross-platform and ultrafast toolkit for FASTA/Q file manipulation. *PLoS One*, **11**, e0163962.

Singer, J. *et al.* (2018) NGS-pipe: a flexible, easily extendable and highly configurable framework for NGS analysis. *Bioinformatics*, **34**, 107–108.

Tilk, S. *et al.* (2019) Accurate allele frequencies from ultra-low coverage pool-seq samples in evolve-and-resequence experiments. *G3 Genes Genomes Genetics*, **9**, 4159–4168.

Vasimuddin, M. *et al.* (2019) Efficient architecture-aware acceleration of BWA-MEM for multicore systems. In: *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. pp. 314–324. https://ieeexplore.ieee.org/document/8820962.

Xu, C. (2018) A review of somatic single nucleotide variant calling algorithms for next-generation sequencing data. *Comput. Struct. Biotechnol. J.*, **16**, 15–24.