Genome analysis

Mapache: a flexible pipeline to map ancient DNA

Samuel Neuenschwander 1,2,†,* , Diana I. Cruz Dávalos 1,3,1, Lucas Anchieri 1,3, Bárbara Sousa da Mota 1,3, Davide Bozzi 1,3, Simone Rubinacci 1,3, Olivier Delaneau 1,3, Simon Rasmussen 1,3, and Anna-Sapfo Malaspinas 1,3

1Department of Computational Biology, University of Lausanne, Lausanne 1015, Switzerland, 2Vital-IT, Swiss Institute of Bioinformatics, Lausanne 1015, Switzerland, 3Swiss Institute of Bioinformatics, Lausanne 1015, Switzerland and 4Novo Nordisk Foundation Center for Protein Research, University of Copenhagen, Copenhagen 2200, Denmark

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
*To whom correspondence should be addressed.

Received on September 23, 2022; revised on December 14, 2022; editorial decision on January 10, 2023; accepted on January 12, 2023

Abstract

Summary: We introduce mapache, a flexible, robust and scalable pipeline to map, quantify and impute ancient and present-day DNA in a reproducible way. Mapache is implemented in the workflow manager Snakemake and is optimized for low-space consumption, allowing to efficiently (re)map large datasets—such as reference panels and multiple extracts and libraries per sample — to one or several genomes. Mapache can easily be customized or combined with other Snakemake tools.

Availability and implementation: Mapache is freely available on GitHub (https://github.com/sneuensc/mapache). An extensive manual is provided at https://github.com/sneuensc/mapache/wiki.

Contact: samuel.neuenschwander@unil.ch

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Mapping sequencing reads to a reference genome is a fundamental step in genomic analyses. Compared with modern data, ancient DNA (aDNA) presents a number of challenges as data are sparse, contaminated and affected by post-mortem damage, such as fragmentation and deamination. Furthermore, samples are often resequenced many times to maximize their yield, making the mapping step iterative. As a result, robust, efficient, yet easy to use bioinformatic pipelines are needed to map and remap the sequencing reads in a reproducible way, while allowing a wide range of users to modify predefined options and navigate through the results to, for example, select promising samples, extracts or libraries. Furthermore, intermediate files might require one order of magnitude more of storage compared with the final BAM files, thus researchers need efficient pipelines since storage is costly and space quotas can be quickly reached during mapping. For example, mapping 8.5 TB of FASTQ files (Harris et al., 2018) would require about 140 TB of storage for temporary files, while the final mapped files measure only 9.9 TB.

While PALEOMIX (based on Python; Schubert et al., 2014) and nf-core/eager (based on Nextflow/Groovy; Yates et al., 2021) have emerged as the main pipelines to map aDNA, both require considerable amounts of storage for temporary files, may not be executed on modern clusters with queuing systems (PALEOMIX) or are limited to a single reference genome (nf-core/eager).

Here, we introduce mapache, a lightweight pipeline designed to map both ancient and present-day DNA sequences to one or multiple reference genomes while generating reports with summary statistics to inspect the quality of the mappings. Mapache is implemented in the workflow manager Snakemake (Molder et al., 2021) and it thus inherits its flexibility, scalability and reproducibility. Snakemake is based on Python allowing users to easily adapt the code or to combine the workflow with other Snakemake workflows. Moreover, unlike the other tools, mapache efficiently manages temporary files, keeping the necessary storage space small. Furthermore, mapache can map sequences to more than one genome in parallel and the pipeline can be used on modern clusters with queuing systems. Finally, mapache has an additional module to allow for genotype imputation of low-coverage genomes with GLIMPSE (Rubinacci et al., 2021), a step that may become quite common in aDNA (see e.g. Sousa da Mota et al., 2022).

2 Approach and features

2.1 Scalability

Mapache scales well for different research cases: screening for aDNA quality (mapping of many small FASTQ files), mapping of high-coverage genomes (many large FASTQ files) to a single reference as well as reconstructing microbial genomes from a metagenomic sample (mapping many FASTQ files to many reference
the input FASTQ files and their metadata (e.g. sample and library names). The amount of memory and running time needed for each step can be specified through a configuration file, as well as parameters specific to the bioinformatic tools included in the pipeline. Mapache can then be launched with a single command line.

As mentioned above, intermediate files can be kept or removed (default) once they are no longer needed. Therefore, if the pipeline fails at any step (e.g. due to a lack of memory or time allocated), it can be re-launched with the same command. This would trigger the re-execution of the failed steps and other pending analyses; that is, successful tasks will not be rerun and their outputs will be automatically used in the next steps when needed.

2.3 Portability
Mapache can easily be installed on any Linux computer. Dependencies are automatically installed using Conda.

3 Workflow

3.1 Configuration file
Mapache can be customized using a configuration file. The configuration is highly flexible, allowing the user to adapt mapache to the input data: (1) steps of the workflow can be skipped, (2) several softwares are available for given steps and (3) any tool-specific parameters may be included in the pipeline. For example, the mapping pipeline can be customized with parameters suited to map present-day DNA; the user can select among BWA-aln, BWA-mem and bowtie2 for mapping; any valid parameters can be passed to the individual tools included in the pipeline. Finally, the workflow can be adapted to the available computer infrastructure by specifying the number of threads, amount of memory and running time for each step in the configuration file.

3.2 Mapping workflow
A detailed description of all the steps of the workflow can be found on the wiki. A diagram of the main steps is shown in Figure 2. Mapache is optimized for mapping aDNA reads to the human reference genome. In brief, adapters, short reads and low-quality bases can be removed with AdapterRemoval2 (Schubert et al., 2016, run by default) or fastp (Chen et al., 2018). Reads are mapped with BWA aln or mem (Li and Durbin, 2009, aln run by default) or with Bowtie2 (Langmead and Salzberg, 2012) and merged by library. Duplicates can be removed per library with Picard MarkDuplicates (https://broadinstitute.github.io/picard/, run by default) or dedup (Peltzer et al., 2016), and libraries are merged by sample. Finally, reads can be realigned locally around indels with GATK (DePristo et al., 2011) and the MD flag can be recomputed with SAMtools (Danecek et al., 2011). Unmapped and low-quality mapped reads can be stored in a separate BAM file per sample.

3.3 Imputation (optional)
Mapache allows for imputation and phasing using GLIMPSE, a tool that is shown to accurately impute both low-coverage present-day (Rubinacci et al., 2021) and ancient human genomes (Sousa da Mota et al., 2022).

3.4 Output
Mapache’s main outputs are the alignments to the reference genomes as BAM files and, if requested, (a) low-quality alignments and unmapped reads and (b) imputed genotypes in VCF format. Mapache outputs summary statistics on the final BAM and intermediate files, including depth of coverage for all or a subset of the chromosomes (Quinlan and Hall, 2010), genomic sex, damage patterns (Jonsson et al., 2013; Malaspinas et al., 2014) and read lengths. Statistics are returned as tables and graphs, allowing to quickly investigate the results. For in-depth analyses, mapache can generate several individual reports (fastqc, Andrews, 2010b; qualitymap, Okonechnikov et al., 2016; multiqc, Ewels et al., 2016). At the end of a run, mapache will output a plain-text file with the versions of the software used. A self-contained html report with all graphs

![Figure 2](https://example.com/figure2.png)
and tables can be generated in a single command at the end of a run. The summary statistics computation and the imputation can also be run for any BAM file not generated by mapache.

4 Conclusion

Mapache is a flexible mapping workflow optimized for runtime and minimal storage usage to efficiently map both ancient and present-day DNA sequences in a reproducible manner. Mapache comes with an extensive wiki and pre-defined parameters to facilitate its use. Furthermore, as the underlying language is Python, users may modify the code or combine mapache with any other Snakemake workflow. Finally, mapache includes downstream processes that are also standalone and can be run on existing BAM files.

Acknowledgements

We thank F Michaud and YO Arizmendi Cárdenas for helpful discussion.

Funding

The work was funded by an SNSF grant (PCEGP3_181251) and an ERC-STG grant (CAMERA 679330) to ASM. SR was supported by the Novo Nordisk Foundation (grant NNF14CC0001).

Conflict of Interest

None declared.

References

Andrews, S. (2010) FASTQC: A quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc.

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

Clemente, F. et al. (2021) The genomic history of the Aegean palatial civilizations. Cell, 184, 2565–2586.e21.

Danecek, P. et al. (2011) Twelve years of SAMtools and BCFtools. Gigascience, 10, 1–4

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.

Harris, D.N. et al. (2018) Evolutionary genomic dynamics of Peruvians before, during, and after the INCA empire. Proc. Natl. Acad. Sci. USA, 115, E6526–E6533.

Jönsson, H. et al. (2013) MapDamage2.0: Fast approximate Bayesian estimates of ancient DNA damage parameters. Bioinformatics, 29, 1682–1684.

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

Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics, 25, 1734–1740.

Malaspinas, A.-S. et al. (2014) Bammds: A tool for assessing the ancestry of low-depth whole-genome data using multidimensional scaling (MDS). Bioinformatics, 30, 2962–2964.

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

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

Pelzner, A. et al. (2016) EAGER: Efficient ancient genome reconstruction. Genome Biol., 17, 60.

Quinlan, A.R. and Hall, I.M. (2010) BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics (Oxford, England), 26, 841–842.

Rubinacci, S. et al. (2021) Efficient phasing and imputation of low-coverage sequencing data using large reference panels. Nat. Genet., 53, 120–126.

Schubert, M. et al. (2014) Characterization of ancient and modern genomes by SNP detection and phylogenomic and metagenomic analysis using PALEOMIX. Nat. Protoc., 9, 1056–1082.

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

Sousa da Mota, B. et al. (2022) Imputation of ancient genomes. bioRxiv. page 2022.07.19.500636.

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

Fig. 2. Main workflow of mapache. Each box shows a step and related software. Dashed/white boxes are optional.