Optimization of nf-core/sarek for large-scale analysis of short-read DNA sequencing data on multiple compute infrastructures

Friederike Hanssen[1], Maxime Garcia[2], Gisela Gabernet[1], Sven Nahnsen[1]
[1]Quantitative Biology Center(QBiC), University of Tübingen [2]SciLifeLab, Karolinska Institutet, Stockholm

1. Introduction

- Somatic variant calling studies often include many patients with dataset sizes varying widely between oncopanels, WXS, and WGS data.
- nf-core[1] provides reproducible, scalable, and portable open-source Nextflow[2]-based pipelines.
- nf-core/sarek[3] is suited for SNP, SV, and CNA calling of tumor/normal paired short-read data including WGS, WXS, and oncopanels
- Local datasets often need to be analyzed on-site due to data security concerns.
- Many public cancer databases are available in commercial clouds. Their analysis can support and enhance local data. However, the download can be time consuming. Processing data in commercial clouds is an alternative, although it can be cost-intensive. (**Fig. 1**)

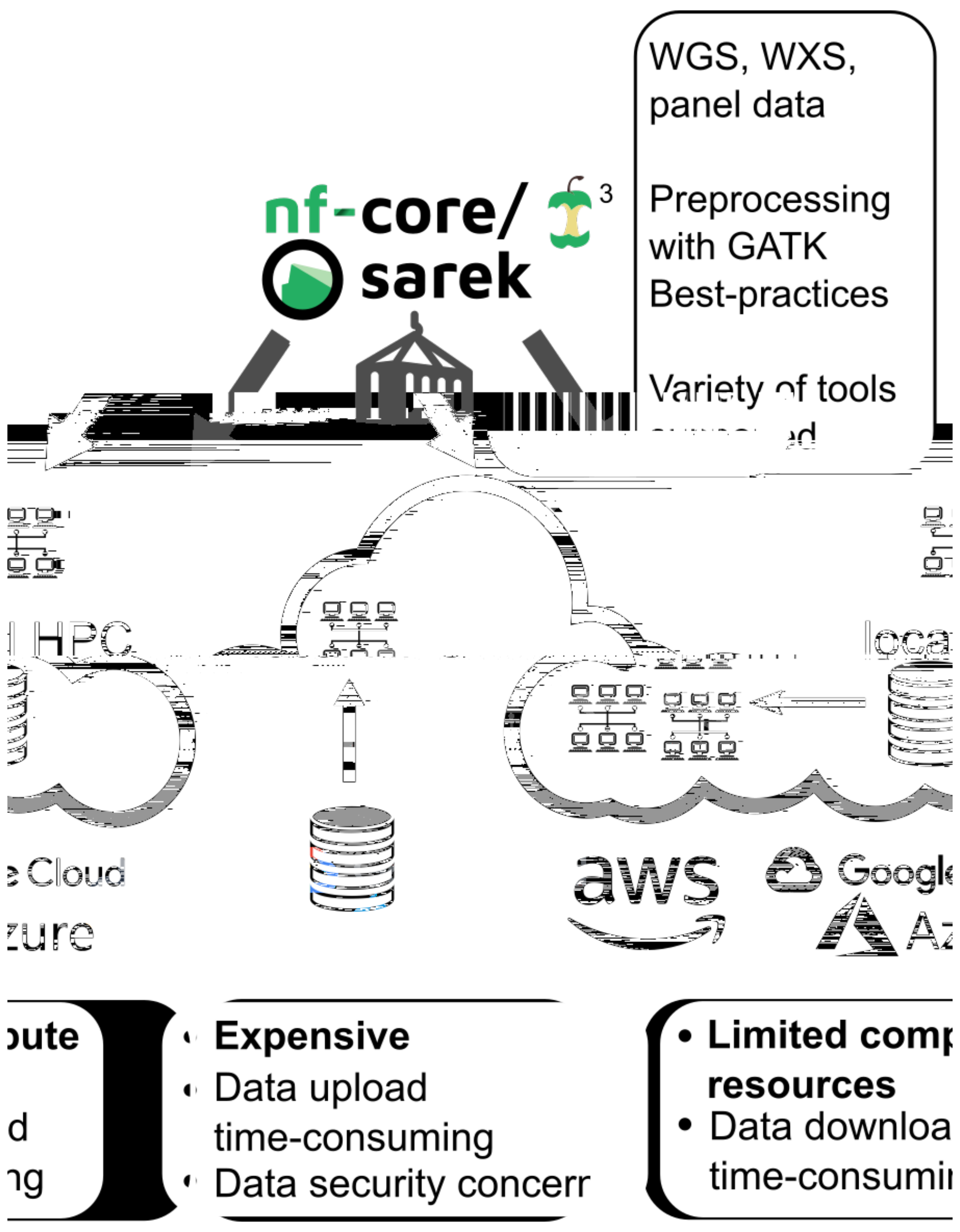

Fig.1: nf-core pipelines are portable across different infrastructures and containerized to ensure reproducible analyses. Processing data in the cloud can be expensive, local infrastructures provide limited resources.

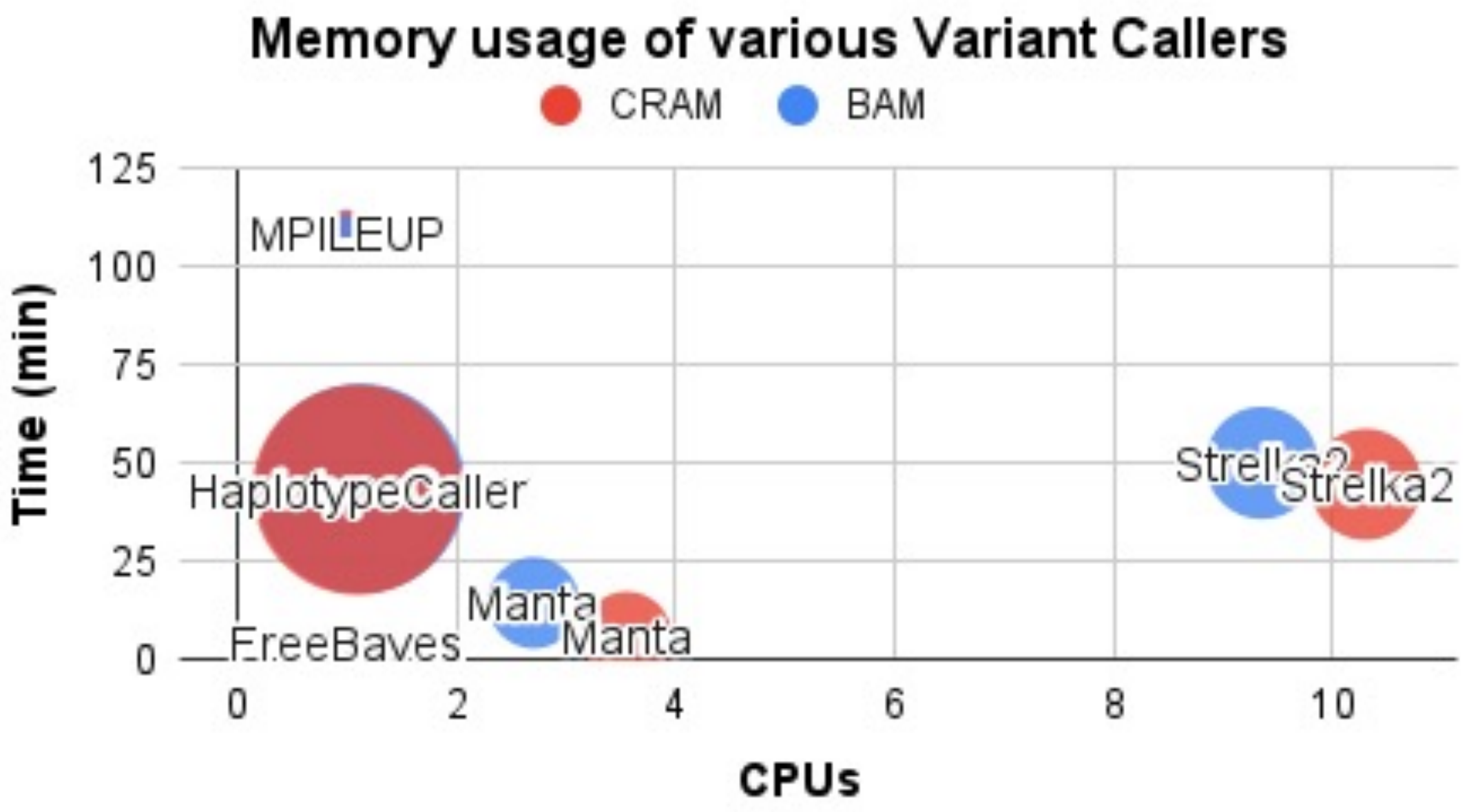

Fig.3: Comparison of selected variant callers included in Sarek either with BAM(blue) or CRAM(red) files as input

2. Methods

Pipeline improvements (Fig. 2)

- Replacement of BAM with CRAM as file:
  40-70% space reduction[4]
- Improving data flow:
  Splitting large input files
  Use implicit file merging by GATK
- Replace & add new tools:
  BWAMem2
  GATK4 Spark
  New variant callers
- Porting to Nextflow DSL2

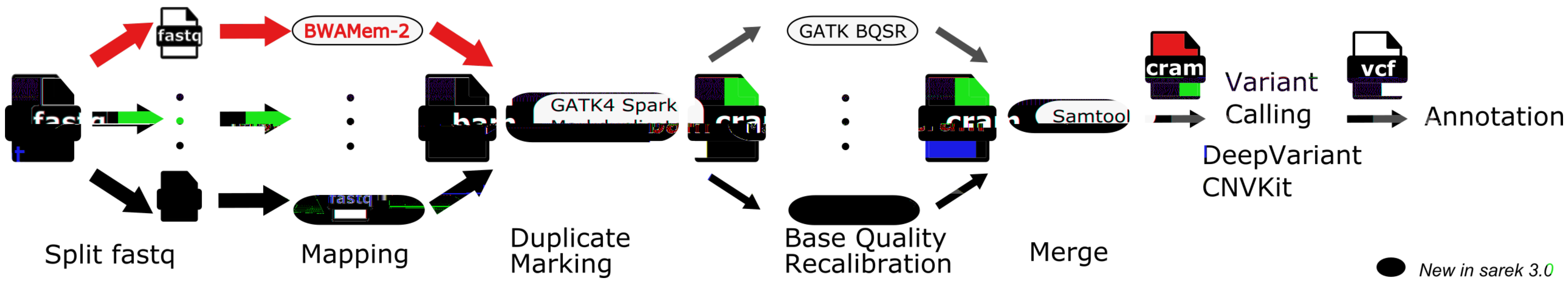

Fig.2: Overview of the changes in nf-core/sarek 3.0

3. Results

- Switching BAM format with CRAM does not increase CPU, memory requirements per machine, or runtime (**Fig. 3**). In addition, replacing the file format with CRAM is in concordance with GH4GA recommendations[4].
- Splitting the input speeds up runtime (**Fig. 4**)
- Porting to DSL2 adheres to nf-core community standards, improving code quality & readability, and facilitating long-term maintenance. The resulting modular implementation can ease custom downstream analysis.
- Costs for runs on AWS setup with Nextflow Tower are reduced in comparison to a naïve, manual setup of the batch environment (**Fig. 5**).

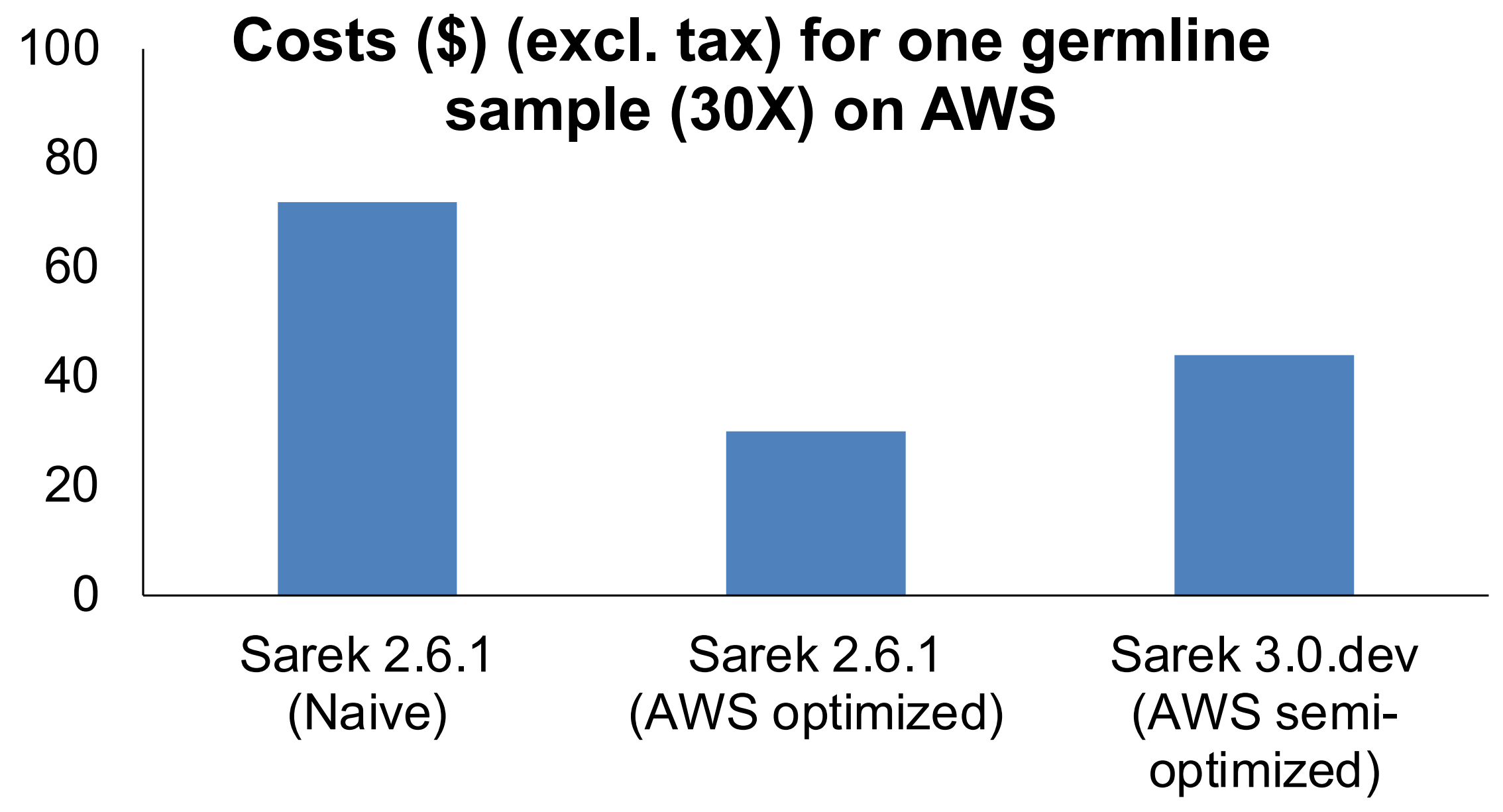

Fig.5: Cost for a single patient (30X, germline) on AWS with a recent release and the new developments.

Conclusion & Outlook

- Replacing file formats reduces space consumption, while not increasing resource usage. Splitting large input files speeds up pre-processing steps.
- Tailor AWS setup & requested resources to new workflow and input data size to further reduce costs.
- Evaluate other commercial cloud providers
- Finish porting and releasing the pipeline. If you want to get involved contact us on Slack or GitHub.

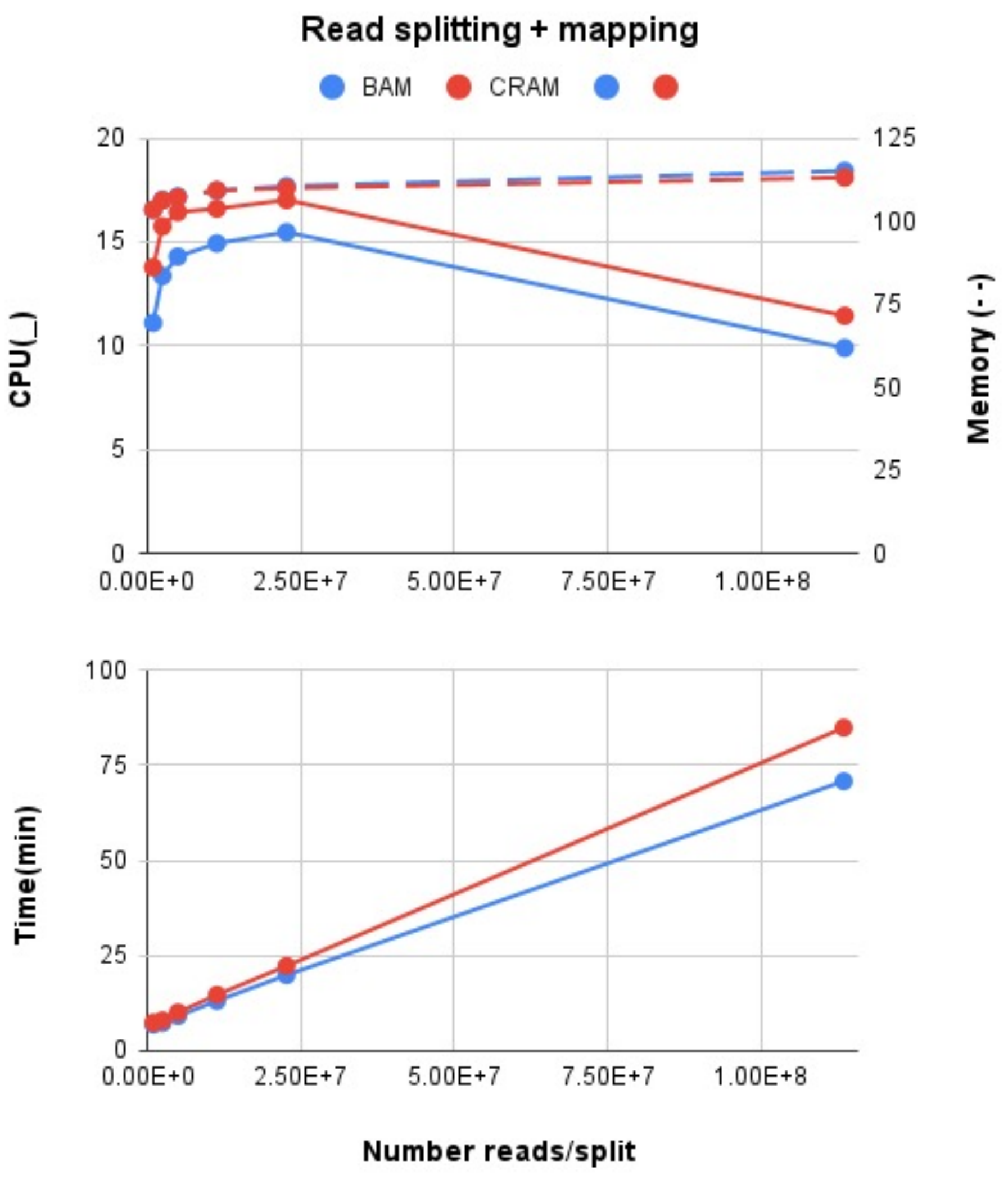

Fig.4: Resource usage of the processing steps read splitting and mapping combined for both BAM and CRAM files as output.

Join us

https://nf-co.re/sarek

GitHub

slack

References

1. Ewels et al. (2020), Nature Biotechnology 38, 276–278
2. Di Tommaso et al. (2017), Nature Biotechnology, 35(4), 316–319
3. Garcia et al. (2020), F1000Research 9:63
4. https://www.ga4gh.org/cram/

Acknowledgements

We would like to acknowledge funding from iFIT, the SFB 209 and support from Amazon Web Services for cloud computing.