SOFTWARE TOOL ARTICLE

Sarek: A portable workflow for whole-genome sequencing analysis of germline and somatic variants [version 1; peer review: 2 approved]

Maxime Garcia1*, Szilveszter Juhos1-3*, Malin Larsson4, Pall I. Olason3, Marcel Martin5, Jesper Eisfeldt6, Sebastian DiLorenzo7, Johanna Sandgren1, Teresita Díaz De Ståhl1, Philip Ewels2, Valtteri Wirta8, Monica Nistér1, Max Käller9, Björn Nystedt13

1Department of Oncology-Pathology, Karolinska Institutet, J5:30 BioClinicum, Visionsgatan 4, Karolinska University Hospital at Solna, Solna, 17164, Sweden
2Department of Biochemistry and Biophysics, Science for Life Laboratory, Stockholm University, Box 1031, Solna, 17121, Sweden
3Department of Cell and Molecular Biology, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Uppsala University, Husargatan 3, Uppsala, 752 37, Sweden
4Department of Physics, Chemistry and Biology, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Linköping University, Linköping, 58183, Sweden
5Department of Biochemistry and Biophysics, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Stockholm University, Box 1031, Solna, 17121, Sweden
6Clinical Genetics, Department of Molecular Medicine and Surgery, Karolinska Institutet, MMK L1:00, Karolinska University Hospital at Solna, Stockholm, 171 76, Sweden
7Department of Medical Sciences, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Uppsala University, Husargatan 3, Uppsala, 752 37, Sweden
8Department of Microbiology, Tumor and Cell Biology, Clinical Genomics Facility, Science for Life Laboratory, Karolinska Institutet, Box 1031, Solna, 171 21, Sweden
9School of Engineering Sciences in Chemistry, Biotechnology and Health, Science for Life Laboratory, KTH Royal Institute of Technology, Box 1031, Solna, 17121, Sweden

* Equal contributors

First published: 29 Jan 2020, 9:63
https://doi.org/10.12688/f1000research.16665.1
Latest published: 04 Sep 2020, 9:63
https://doi.org/10.12688/f1000research.16665.2

Abstract
Whole-genome sequencing (WGS) is a fundamental technology for research to advance precision medicine, but the limited availability of portable and user-friendly workflows for WGS analyses poses a major challenge for many research groups and hampers scientific progress. Here we present Sarek, an open-source workflow to detect germline variants and somatic mutations based on sequencing data from WGS, whole-exome sequencing (WES), or gene panels. Sarek features (i) easy installation, (ii) robust portability across different computer environments, (iii) comprehensive documentation, (iv) transparent and easy-to-read code, and (v) extensive quality metrics reporting.
Sarek is implemented in the Nextflow workflow language and supports both Docker and Singularity containers as well as Conda environments, making it ideal for easy deployment on any POSIX-compatible computers and cloud compute environments. Sarek follows the GATK best-practice recommendations for read alignment and pre-processing, and includes a wide range of software for the identification and annotation of germline and somatic single-nucleotide variants, insertion and deletion variants, structural variants, tumour sample purity, and variations in ploidy and copy number. Sarek offers easy, efficient, and reproducible WGS analyses, and can readily be used both as a production workflow at sequencing facilities and as a powerful stand-alone tool for individual research groups. The Sarek source code, documentation and installation instructions are freely available at https://github.com/nf-core/sarek and at https://nf-co.re/sarek/.

Keywords
Analysis workflow, Whole Genome Sequencing, Germline variants, Somatic variants, Cancer
Introduction

Whole-genome sequencing (WGS) and whole-exome sequencing (WES) technologies open up new avenues for research and for clinical applications, with many large initiatives launched worldwide. While much effort has been invested in novel sequencing analysis software, the importance of providing and maintaining workflows to combine software in an efficient and reproducible manner has been underestimated and too few resources are typically dedicated to address this issue. This is of particular importance for somatic variant analysis and especially for analysis of complex cancer genomes, where a combination of tools is still required for optimal sensitivity and specificity and to detect various types of gene mutations and other abnormalities (Alioto et al., 2015). Some encouraging solutions have been presented in recent years, including SeqMule (Guo et al., 2015), SpeedSeq (Chiang et al., 2015), Bcbio-nextgen, and DNAp (Causey et al., 2018). While all of the above represent commendable and important efforts, we have not found any workflow solution that in our opinion fulfills all of the following important user aspects: (i) easy installation, (ii) robust portability across different compute environments, (iii) comprehensive documentation, (iv) transparent and easy-to-read code, and (v) extensive quality metrics reporting. Here we present Sarek, an easy-to-install community-maintained workflow, offering a complete and scalable solution for germline and somatic variant detection, annotation and quality control. Sarek supports several reference genomes and can handle data from WGS, WES and gene panels, and is intended to be used both as a production workflow at core facilities and as a standalone tool for individual research groups. By using Docker or Singularity containers, Sarek installs easily on all POSIX compatible systems such as Linux and Mac OS X and is designed to work on compute environments dedicated to handle sensitive personal data without direct internet access—a situation expected to become increasingly common with growing data security awareness.

Methods

Operation: Workflow overview and software

Sarek offers a portable workflow for germline and somatic variant detection, annotation and quality control based on WGS, WES or gene panel data, using a range of state-of-the-art software and data resources in the field (Table 1, Figure 1). In the pre-processing step, sequence reads are aligned to the reference genome and marked as duplicates. This information is then used to filter out any alignment artifacts, which are then used for downstream analysis. The workflow is designed to be easy to install and to work on different types of compute environments. The software requirements and implementation details are described in Table 1.

Table 1. Software required and implemented in Sarek. A list of all the software required and currently implemented in Sarek. All analysis and quality metrics software are installed automatically when Sarek is launched.

<table>
<thead>
<tr>
<th>Software/Resource</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required software</strong></td>
<td></td>
</tr>
<tr>
<td>Nextflow</td>
<td><a href="https://www.nextflow.io/index.html">https://www.nextflow.io/index.html</a></td>
</tr>
<tr>
<td><strong>Included analysis software</strong></td>
<td></td>
</tr>
<tr>
<td>BWA-MEM</td>
<td><a href="http://bio-bwa.sourceforge.net/">http://bio-bwa.sourceforge.net/</a></td>
</tr>
<tr>
<td>GATK</td>
<td><a href="https://software.broadinstitute.org/gatk/">https://software.broadinstitute.org/gatk/</a></td>
</tr>
<tr>
<td>Samtools</td>
<td><a href="https://github.com/samtools/samtools">https://github.com/samtools/samtools</a></td>
</tr>
<tr>
<td>Strelka2</td>
<td><a href="https://github.com/Illumina/strelka">https://github.com/Illumina/strelka</a></td>
</tr>
<tr>
<td>Manta</td>
<td><a href="https://github.com/Illumina/manta">https://github.com/Illumina/manta</a></td>
</tr>
<tr>
<td>TIDDIT</td>
<td><a href="https://github.com/SciLifeLab/TIDDIT">https://github.com/SciLifeLab/TIDDIT</a></td>
</tr>
<tr>
<td>ASCAT</td>
<td><a href="https://github.com/Crick-CancerGenomics/ascat">https://github.com/Crick-CancerGenomics/ascat</a></td>
</tr>
<tr>
<td>Control-FREEC</td>
<td><a href="http://boevalab.inf.ethz.ch/FREEC/">http://boevalab.inf.ethz.ch/FREEC/</a></td>
</tr>
<tr>
<td>snpEff</td>
<td><a href="http://snpeff.sourceforge.net/">http://snpeff.sourceforge.net/</a></td>
</tr>
<tr>
<td>VEP</td>
<td><a href="http://www.ensembl.org/vep">http://www.ensembl.org/vep</a></td>
</tr>
<tr>
<td><strong>Included quality metrics software</strong></td>
<td></td>
</tr>
<tr>
<td>MultiQC</td>
<td><a href="http://multiqc.info/">http://multiqc.info/</a></td>
</tr>
<tr>
<td>FastQC</td>
<td><a href="https://www.bioinformatics.babraham.ac.uk/projects/fastqc/">https://www.bioinformatics.babraham.ac.uk/projects/fastqc/</a></td>
</tr>
<tr>
<td>BamQC</td>
<td><a href="https://github.com/s-andrews/BamQC">https://github.com/s-andrews/BamQC</a></td>
</tr>
<tr>
<td>QualiMap</td>
<td><a href="http://qualimap.bioinfo.cipf.es/">http://qualimap.bioinfo.cipf.es/</a></td>
</tr>
<tr>
<td>BCFtools</td>
<td><a href="https://github.com/samtools/bcftools">https://github.com/samtools/bcftools</a></td>
</tr>
<tr>
<td>VCFtools</td>
<td><a href="https://vcftools.github.io/index.html">https://vcftools.github.io/index.html</a></td>
</tr>
</tbody>
</table>
Figure 1. Schematic overview of the Sarek workflow for analysis of germline and somatic variants. A schematic overview including some of the main analysis software implemented in the Sarek workflow. A more comprehensive list of the currently implemented software is given in Table 1.

Portability and reproducibility
Sarek is implemented in Nextflow (Di Tommaso et al. 2017), a workflow language designed specifically for bioinformatics applications. Nextflow has a transparent design, making the Sarek code easy to read, adjust and extend. Compared to the Bpipe workflow language (used in for example DNAp), Nextflow offers superior support for different execution environments, like Slurm, Sun Grid Engine, LSF and Kubernetes, and includes native support for cloud compute environments including Google Cloud and AWS. Support for AWS batch gives the possibility to easily distribute thousands of batch jobs on Amazon Web Services. Sarek is part of a rapidly growing community effort of well documented and community-tested Nextflow pipelines, and adheres to the nf-core portability and documentation guidelines (Ewels et al., 2019). To facilitate easy installation and to ensure reproducibility, all Sarek required tools are managed in Docker or Singularity (Kurtzer et al., 2017) containers, or a Conda environment. While Docker is a widely appreciated container solution, it is not always allowed at high-performance computing centers because of the involved security risks, making Singularity the preferred choice at these sites (Kurtzer et al., 2017). This is of particular importance for computer environments designed for handling sensitive personal data, where a high level of data security has to be maintained across multiple projects and users.

Implementation: Equipment and resource usage
Sarek can be installed and executed on any POSIX-compatible computer system. To run a full WGS analysis, including both germline and somatic variants from a tumour/normal dataset with 90x/90x read coverage, we recommend a minimum of 16 cores on a node with 128 GB RAM, and at least 4 TB available free storage (in addition to the initial FASTQ files) in the input/output working directory. Of this, about 1.4 TB will be allocated for BAM files, annotated VCF files and CNV files, but excluding GVCF files (Table 2). At the end of the run, 2.3 TB temporary data can be removed, unless the user plans to perform re-runs from intermediate processing states. Many processes are distributed across cores by dividing the genome into smaller chunks, each being handled as a separate core job, with all the results being merged and sorted in a final step. Some of the used software are parallelized by design, while for others Sarek uses a scatter-gather approach to efficiently distribute the processing load across CPU cores and reduce the wall clock runtime.

Installation and testing
Sarek is run from a computer system with a local installation of Nextflow and support for either Conda environments,
Docker or Singularity containers. Nextflow can automatically fetch the Sarek source code from GitHub. All software dependencies are encapsulated in Docker or Singularity containers which are downloaded from Docker Hub, or built in a new Conda environment using Bioconda (Grüning et al., 2018). As such, cumbersome software installations by the user are completely avoided. Configuration files allow tailoring to specific user needs and incomplete runs are easily restarted from any stage in the workflow process. Sarek comes with a small test data set and a suite of tests to verify the installation. This is also used for Continuous Integration testing with GitHub Actions.

Results
To test performance in terms of resource usage and biological results, Sarek was run on a medulloblastoma WGS tumour/normal dataset from a sample with high tumour cell content (~98%), and with a curated “Gold Set” of verified somatic mutations from a previous benchmark study (Alioto et al., 2015). In line with the above benchmark study, Sarek (version 2.5.2) was executed with WGS germline and somatic variant calling using a 90X/90X tumour/normal dataset (accession number EGAD00001001859, read sets EGAR00001387019-24 and EGAR00001387025-32). Runs were performed on a single 48-thread node with a local direct attached storage (DAS): A Dell PowerEdge R740 server, with two Intel Xeon Gold 6126 with a total of 24 cores (48 threads) CPUs, 756 GB memory, and 100 TB SCv3020 Compellent Storage. The complete Sarek run including preprocessing followed by both germline and somatic variant calling and annotation took 48 hours and 21 minutes, and required about three times more storage than the original input data (Table 2). Notably, the complete Sarek run was executed by a single command, with fully automated installation, execution, and efficient job distributions of the more than 15 different software tools to complete the analysis and provide quality control metrics, without any manual intervention needed during the two-day run. To ensure that the Sarek output was biologically sound, we calculated precision, recall and F1 statistics for the Sarek output based on the “Gold Set” of somatic single-base mutations (SSM) and somatic insertion/deletion mutations (SIM) as previously defined (Alioto et al., 2015). Using the intersection of the output from the two somatic variant callers (GATK4 Mutect2 and Strelka2), Sarek provided accuracy measures for SSMs (F1 score = 0.80) and SIMs (F1 score = 0.58) in the top range of the 18 somatic variant calling procedures included in the original benchmarking study on this data set (Table 3), indicating that the workflow operates as intended. The sample purity was estimated to be 100%, as compared to 98% previously reported for this sample. For somatic structural variants and ploidy, no relevant benchmark data was available, and therefore no quantitative assessment beyond previously published results for the implemented software could be performed, but the integrity of the runs were checked by comparing the results of Manta, ASCAT, and Control-FREEC run within Sarek and as stand-alone.

### Use case
Sarek has been extensively tested and applied on various WGS datasets, including thousands of samples for germline variant analyses, and hundreds of paired tumour/normal samples for somatic mutation analyses. In addition, Sarek has also been successfully tested on WES data and gene panels. Below we present a standard use case with a tumour/normal WGS dataset as input, running both germline and somatic variant analyses.

### Input data
For a somatic variant analysis, the user should provide the sequencing FASTQ files from both tumour and normal control tissue from the same individual, described in a tab-delimited TSV file (here: `samples.tsv`). Each line of the TSV file contains information about a sequence data file, including: The identifier of the individual, the gender (XX or XY), the status of the sample (0 for Normal or 1 for Tumour), the identifier of the sample, the sequencing lane (if samples are multiplexed across multiple lanes), and the paths to the FASTQ file of the first and second read in the read-pair. Relapse samples from the same individual are also supported.

### Running Sarek on WGS data with Singularity containers
Running Sarek with Singularity container on a computer system supporting Java 8 requires only installation of Nextflow and Singularity. A full analysis run starting from FASTQ files including mapping, recalibration, variant calling and annotation,
as well as generating a full QC report can be invoked by a single Nextflow command:

```bash
> nextflow run nf-core/sarek -r 2.5.2 -profile singularity --input samples.tsv --tools Mutect2,Strelka,Manta,TIDDIT,ASCAT,ControlFREEC,snpEff,VEP
```

Nextflow will recognize the workflow name and will download the specified version (2.5.2) of the pipeline from GitHub, including the corresponding container, as well as fetching the required reference files from AWS-iGenome. The default reference genome is human GRCh38, but Sarek also supports GRCh37 and nearly 30 other genomes directly accessible from iGenomes. Alternatively, users can manually supply Sarek with other reference genomes. Non-default parameters and links to local reference files are handled in accordance with nf-core guidelines. User configuration profiles can be stored locally or centrally at https://github.com/nf-core/configs.

### Discussion

Human WGS is transforming medical research, and provides a foundation to develop novel clinical applications and improve health care. An important aspect to harvesting the potential of WGS is however to empower the research community with adequate bioinformatics tools, and reproducible bioinformatics workflows are important drivers of scientific progress by making complex processing of large datasets feasible for a wide range of researchers. While we are highly appreciative of existing workflows for cancer and non-cancer variant detection, we argue that there is no one-size-fits-all solution and more initiatives are needed to serve the large and diverse research user community, especially for WGS data. Sarek builds on a philosophy of reasonably narrow, independent workflows, written in the domain-specific language Nextflow. In our experience, this is an effective strategy to simplify workflow maintenance at sequencing core facilities, and to allow easy deployment and modifications by individual research groups. Sarek efficiently utilizes cloud and high-performance compute clusters and installs easily across compute environments. Sarek provides annotated VCF files, CNV reports and quality metrics for germline and cancer samples from raw FASTQ sequencing data in about 48 hours for 90X/90X WGS data (as demonstrated here), in a few hours for WES data, and within minutes for gene panels (in-house data, not presented here). Ongoing efforts aim to develop add-on ranking and visualization modules and to efficiently extract clinically and biologically relevant findings, to help advance basic and translational research.

### Conclusion

Sarek is a portable and reproducible workflow to detect germline and somatic variants from WGS, WES and gene panel data. It includes extensive analysis and quality control metrics, while still being limited to a relatively narrow scope to
Data availability

Source data

European Genome-phenome Archive: A comprehensive assessment of somatic mutation detection in cancer using whole genome sequencing. https://www.ebi.ac.uk/ega/datasets/EGAD00001001859. Read sets EGAR00001387019-24 and EGAR00001387025-32 were analysed.

These data are held under restricted access. Readers wishing to apply for access to the data must first apply through the ICGC Data Access Compliance Office (https://icgc.org/daco) and complete the data access form. Access will be granted to those whose projects conform to the goals and policies of ICGC. Help with completing the data access form is available at https://icgc.org/daco/help-guide-section.

The workflow itself comes with a prebuilt profile with a complete configuration for automated testing, including links to a small test dataset.

Software availability

Sarek is available at: https://nf-co.re/sarek.

Source code available at: https://github.com/nf-core/sarek.

Archived source code at time of publication: https://doi.org/10.5281/zenodo.3579102 (Garcia et al., 2019).

License: MIT License.

Author contributions

MK, BN and MN conceived the idea for Sarek. MG and SJ led the project. MG, SJ, ML, PIO, MM, JE, and SDL designed and implemented the workflow. JS, TDS, VW, MN, BN, PE and MK performed testing and provided design feedback. MG, SJ and BN wrote the manuscript with the help from all authors.

Acknowledgements

We are grateful for the valuable input from the Oslo University Hospital bioinformatics core facility (Oslo University Hospital), the T Martinsson lab (Gothenburg University), the A–C Syvänen lab (Uppsala University), and Alex Peltzer (Quantitative Biology Center, University of Tübingen). The National Genomics Infrastructure (NGI) and Uppsala Multidisciplinary Centre for Advanced Computational Science (UPPMAX) provided computational resources. Help with graphical design was provided by Dr. Jonas Söderberg (Uppsala university).

References

Li H: Aligning sequence reads, clone sequences and assembly contigs with
Reference Source
PubMed Abstract | Publisher Full Text | Free Full Text
PubMed Abstract | Publisher Full Text | Free Full Text
PubMed Abstract | Publisher Full Text | Free Full Text
PubMed Abstract | Publisher Full Text | Free Full Text
PubMed Abstract | Publisher Full Text | Free Full Text
Open Peer Review

Current Peer Review Status: ✔️ ✔️

Version 1

Reviewer Report 31 March 2020

https://doi.org/10.5256/f1000research.18214.r61129

© 2020 Pitkänen E. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✅ Esa Pitkänen

1 Institute for Molecular Medicine Finland (FIMM), Helsinki Institute of Life Science, University of Helsinki, Helsinki, Finland
2 Applied Tumor Genomics Research Program, Faculty of Medicine, University of Helsinki, Helsinki, Finland

This manuscript describes Sarek, a workflow for analyzing next-generation sequencing (NGS) data. Sarek is based on Nextflow, a popular tool for defining computational workflows. In order to process NGS data, i.e., generating annotated variant calls ready for downstream analyses, multiple complex software tools need to be executed. This is not only computationally demanding, but also labor-intensive due to operators having to install and maintain a complicated collection of software as well as diagnose failed analysis runs, often resulting in high management overhead compared to the total computation time (Yakneen and Waszak, 2020¹). Sarek aims to minimize the installation and management time overhead by building a NGS workflow on top of Nextflow, automatically installing the required software components. These software consist of some of the state-of-the-art tools in read mapping, variant calling and annotation and quality control. Sarek is a welcome addition to the toolkit of bioinformaticians looking for an NGS analysis workflow, which can be easily installed on a HPC cluster or cloud environment. The article is well-written and clear to understand. While I'm happy to recommend indexing of the manuscript also in the present form, I have a few suggestions how to improve it:

**Major comments:**

1. While some of the existing NGS workflows are mentioned, I would appreciate it if Sarek was compared to these approaches in more detail. Is there functionality that is currently missing from Sarek that is present in one of the other workflows?

2. Typically in NGS data analysis, a lot of time can be spent on debugging failed runs to find out whether one of the tools failed, if the data is corrupted/missing or if there was a hardware error. How does Sarek support run diagnostics and relaunching failed jobs?

3. How does Sarek combine variant calls when multiple callers are used for a variant type?
4. Somatic single nucleotide and indel variant calls from Sarek were shown to match well with a previously defined gold standard callset. No benchmark data was available for more complex somatic variants and variant calling accuracy for germline variants was not evaluated. I am interested in seeing more comprehensive tests to cover all germline and somatic variant types.

5. I would appreciate it if a minimal test dataset together with instructions and a suite of automated tests was provided with Sarek. This would make it easier for the user to test out an installation as well as raising issues in GitHub.

Minor comments:
1. How easy it is for users to modify or extend Sarek by for example adding a new variant caller to the workflow? This could be explored in more detail in text.

2. It would be good to easily see which tools are used to call and analyze each variant type. This information could be added either to Fig 1, Table 1 or both.

3. FreeBayes is included in Figure 1 but missing from Table 1.

4. The wording in “To facilitate easy installation and to ensure reproducibility, all Sarek required tools are managed in Docker or Singularity (Kurtzer et al., 2017) containers, or a Conda environment.” should be clarified -- are all tools being maintained in all the three systems?

5. When running Sarek with default options, it crashes in the tool version check. It took me a while to figure out this was due to the default “-profile” argument of “standard” which seems to assume Singularity is available. It would be good to improve error messages so that it is easier to understand the underlying cause. A minimal installation and testing procedure mentioned above would help in this regard.

6. Typo: “Whole-genome sequencing (WGS) and whole-exome sequencing (WES) technologies opens...” -> “...open”.

References
PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

**Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?**
Yes

**Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?**
Yes

**Competing Interests:** No competing interests were disclosed.

**Reviewer Expertise:** Bioinformatics, cancer genetics, machine learning

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

---

Author Response 01 Jul 2020

**Björn Nystedt,** National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Uppsala University, Husargatan 3, Uppsala, Sweden

We are grateful to the reviewer for the positive and constructive comments! We have uploaded a revised version of the manuscript and included updated documentation for the workflow, including adjustments and improvements as suggested by the reviewer, and as described in the detailed comments below, marked in bold.

Approved

This manuscript describes Sarek, a workflow for analyzing next-generation sequencing (NGS) data. Sarek is based on Nextflow, a popular tool for defining computational workflows. In order to process NGS data, i.e., generating annotated variant calls ready for downstream analyses, multiple complex software tools need to be executed. This is not only computationally demanding, but also labor-intensive due to operators having to install and maintain a complicated collection of software as well as diagnose failed analysis runs, often resulting in high management overhead compared to the total computation time (Yakneen and Waszak, 2020). Sarek aims to minimize the installation and management time overhead by building a NGS workflow on top of Nextflow, automatically installing the required software components. These software consist of some of the state-of-the-art tools in read mapping, variant calling and annotation and quality control. Sarek is a welcome addition to the toolkit of bioinformaticians looking for an NGS analysis workflow, which can be easily installed on a HPC cluster or cloud environment. The article is well-written and clear to understand. While I'm happy to recommend indexing of the manuscript also in the present form, I have a few suggestions how to improve it:

**Major comments:**

1. While some of the existing NGS workflows are mentioned, I would appreciate it if
Sarek was compared to these approaches in more detail. Is there functionality that is currently missing from Sarek that is present in one of the other workflows?

This is a relevant comment, but a detailed comparison of the current functionality of different workflows risk being quickly outdated as functionality will frequently change both in Sarek and in the other mentioned workflows. Also, the main purpose of Sarek is not to provide unique functionality *per se*, but to provide a workflow solution with some generally important features as detailed in the manuscript “..(i) easy installation, (ii) robust portability across different compute environments, (iii) comprehensive documentation, (iv) transparent and easy-to-read code, and (v) extensive quality metrics reporting.”

Therefore, the main difference between Sarek and the other mentioned workflows is in practical usability rather than a particular functionality, as these can typically be tuned or added as needed.

1. Typically in NGS data analysis, a lot of time can be spent on debugging failed runs to find out whether one of the tools failed, if the data is corrupted/missing or if there was a hardware error. How does Sarek support run diagnostics and relaunching failed jobs?

This is a good comment, and NextFlow reports the failed process and the error-code from the underlying software, and Sarek is designed to make it very easy to resume failed jobs from the point of failure. This has now been better highlighted in the manuscript under the heading “Portability and reproducibility”. We will work continuously with the user community to further avoid run failures and to continuously improve the diagnostic capability and error-handling.

1. How does Sarek combine variant calls when multiple callers are used for a variant type?

*Sarek will report variants from all callers include in the run, but it is up to the user to decide on how to combine results from different callers, since e.g. the optimal balance between high specificity versus high sensitivity differs across research projects. This has been clarified in the manuscript under the heading “Output”, and in the “Discussion”.

1. Somatic single nucleotide and indel variant calls from Sarek were shown to match well with a previously defined gold standard callset. No benchmark data was available for more complex somatic variants and variant calling accuracy for germline variants was not evaluated. I am interested in seeing more comprehensive tests to cover all germline and somatic variant types.

This is a good suggestion and we have run germline variant calling with Sarek on the HG001 sample and compared to the Genome In a Bottle gold standard dataset, and these results are now included in the manuscript.

Benchmarking of complex somatic variants is very complex and difficult due to the lack of robust and relevant benchmark datasets making such testing beyond the scope of this publications, since Sarek is a workflow and does not provide any novel algorithms or methodology *per se*. For now, we refer users to the tests published along with the respective included variant calling software. This limitation is stated in the manuscript under the heading “Results”.

1. I would appreciate it if a minimal test dataset together with instructions and a suite of automated tests was provided with Sarek. This would make it easier for the user to
test out an installation as well as raising issues in GitHub.

This is a good suggestion, and there is actually already a minimal test dataset and a suit of tests available and documented in Sarek, as detailed under the heading “Installation and testing”. We have improved the Sarek documentation to make this clear and easy to find.

Minor comments:

1. How easy it is for users to modify or extend Sarek by for example adding a new variant caller to the workflow? This could be explored in more detail in text.

This is a good suggestion and we have clarified this point in the manuscript under the heading “Operation: Workflow overview and software”. In brief, we have started with software we have judged being of high quality, well-maintained and robust. Additional software will be added to Sarek later on in a community effort, and this process is already ongoing.

1. It would be good to easily see which tools are used to call and analyze each variant type. This information could be added either to Fig 1, Table 1 or both.

This is a good suggestion and we have added this information to Table 1.

1. FreeBayes is included in Figure 1 but missing from Table 1.

This is a good note, and we have added this information to Table 1.

1. The wording in “To facilitate easy installation and to ensure reproducibility, all Sarek required tools are managed in Docker or Singularity (Kurtzer et al., 2017) containers, or a Conda environment.” should be clarified -- are all tools being maintained in all the three systems?

This is a very useful comment, and this has now been clarified in the manuscript under the heading “Portability and reproducibility”. All tools are installed in Conda, and then pushed to DockerHub (https://hub.docker.com/). This way all tools are available directly from all three systems; Conda, Docker and Singularity.

1. When running Sarek with default options, it crashes in the tool version check. It took me a while to figure out this was due to the default “-profile” argument of “standard” which seems to assume Singularity is available. It would be good to improve error messages so that it is easier to understand the underlying cause. A minimal installation and testing procedure mentioned above would help in this regard.

This is a very useful comment, and we have improved the error message and the documentation regarding the “-profile” arguments.

1. Typo: “Whole-genome sequencing (WGS) and whole-exome sequencing (WES) technologies opens...” -> “…open”.

This typo has been corrected in the manuscript.

- Is the rationale for developing the new software tool clearly explained?
  Yes
- Is the description of the software tool technically sound?
  Yes
Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Report 09 March 2020

https://doi.org/10.5256/f1000research.18214.r59295

© 2020 Håndstad T. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Tony Håndstad
Oslo University Hospital, Oslo, Norway

Sarek is a workflow for variant detection and analysis of sequencing data from WGS, WES and targeted panels. The workflow is comprehensive and versatile, allowing for variant detection in both germline and somatic samples, from WGS/WES/panel sequencing.

- It includes variant calling of SNPs, indels, and structural variants, as well as annotation and extensive quality control.

- Sarek is open source and part of the nf-core community effort which builds well-curated analysis pipelines in the Nextflow pipeline framework.

- Sarek is very user friendly, and installation, configuration and execution is easy to perform, while the workflow is also flexible.

- Implementation manages to be clear, despite also being advanced with parallelization and ample choice of installation/execution. Many researchers would likely struggle to implement pipelines at this advanced level.

- The documentation is excellent, and despite the comprehensive functionality, most users should find it easy to set up Sarek and get started. I have no doubt that Sarek will be a very valuable addition to the research community.

The paper is well written and fulfils all the reviewer criteria. As reviewer, I have only a few minor comments:
o Sarek can use different Nextflow configuration profiles. In the Sarek documentation, it says that the test profile is a profile with a complete configuration for automated testing and that it includes links to test data so needs no other parameters.

It should be obvious to most users, but I would suggest that the authors make it clear that when using the test profile, a user must also supply conda, docker or singularity profile if not all the tools are installed in the PATH. This is clear from the general nf-core documentation (https://nf-co.re/usage/introduction) but less so from the Sarek documentation.

o Whereas the choice of Nextflow is justified, there is little argumentation for why the different tools (variant callers in particular) are selected other than that they represent state-of-the-art. Also why several tools for variant calling are combined is not mentioned, though the referred paper (Alioto et al., 2015) makes a clear case for this, at least for somatic variant calling.

o The paper title makes it clear that Sarek is for whole genome sequencing analysis, but as stated in the text, Sarek is also applicable to exome and targeted panel analyses where the authors say it has been run successfully.

Many researchers are using exome sequencing, so it could be of interest to know if the authors have an opinion or experience with how use of targeted sequencing data limit Sarek in terms of accuracy or utility, for example, are the tools used for structural variant calling able to handle exome data well (to the extent possible with targeted sequencing?)

The documentation has a small chapter stating that the authors recommend supplying a BED file with the targeted regions, but there is not so much explanation of what the effect of this is.

o The authors demonstrate that Sarek is both fast and accurate by running it on a tumor/normal(germline) dataset from a previous benchmark study. I think this is acceptable/sufficient, but one could always wish for more; the paper could be strengthened by for example running the well-known public germline HG001 sample against the Genome In a Bottle gold standard dataset.

o Accurate somatic variant calling is difficult. But the included benchmark study demonstrates that Sarek performs well in comparison with other pipelines. The tool leaves it up to the user to decide whether to use output from a single variant caller or the union or intersection from all tools for increased sensitivity or precision.

o In summary, I think Sarek is a great addition to the community and recommend the paper for indexing.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Diagnostic bioinformatics (variant calling pipelines) and variant interpretation

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 01 Jul 2020

Björn Nystedt, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Uppsala University, Husargatan 3, Uppsala, Sweden

We are grateful to the reviewer for the positive and constructive comments! We have uploaded a revised version of the manuscript and included updated documentation for the workflow, including adjustments and improvements as suggested by the reviewer, and as described in the detailed comments below, marked in bold.

Approved

Sarek is a workflow for variant detection and analysis of sequencing data from WGS, WES and targeted panels. The workflow is comprehensive and versatile, allowing for variant detection in both germline and somatic samples, from WGS/WES/panel sequencing.

- It includes variant calling of SNPs, indels, and structural variants, as well as annotation and extensive quality control.

- Sarek is open source and part of the nf-core community effort which builds well-curated analysis pipelines in the Nextflow pipeline framework.

- Sarek is very user friendly, and installation, configuration and execution is easy to perform, while the workflow is also flexible.

- Implementation manages to be clear, despite also being advanced with parallelization and ample choice of installation/execution. Many researchers would
likely struggle to implement pipelines at this advanced level.

- The documentation is excellent, and despite the comprehensive functionality, most users should find it easy to set up Sarek and get started. I have no doubt that Sarek will be a very valuable addition to the research community.

The paper is well written and fulfils all the reviewer criteria. As reviewer, I have only a few minor comments:

- Sarek can use different Nextflow configuration profiles. In the Sarek documentation, it says that the test profile is a profile with a complete configuration for automated testing and that it includes links to test data so needs no other parameters.

It should be obvious to most users, but I would suggest that the authors make it clear that when using the test profile, a user must also supply conda, docker or singularity profile if not all the tools are installed in the PATH. This is clear from the general nf-core documentation (https://nf-co.re/usage/introduction) but less so from the Sarek documentation.

This is a good suggestion and we have highlighted and improved this information in the Sarek documentation. We are also working to revise the general documentation format in nf-core to make this more transparent throughout.

- Whereas the choice of Nextflow is justified, there is little argumentation for why the different tools (variant callers in particular) are selected other than that they represent state-of-the-art. Also why several tools for variant calling are combined is not mentioned, though the referred paper (Alioto et al., 2015) makes a clear case for this, at least for somatic variant calling.

This is a good suggestion and we have clarified this point in the manuscript. In brief, we have included software we have judged being of high quality, well-maintained and robust. Additional software will be added to Sarek later on in a community effort, and this process is already ongoing.

- The paper title makes it clear that Sarek is for whole genome sequencing analysis, but as stated in the text, Sarek is also applicable to exome and targeted panel analyses where the authors say it has been run successfully.

Many researchers are using exome sequencing, so it could be of interest to know if the authors have an opinion or experience with how use of targeted sequencing data limit Sarek in terms of accuracy or utility, for example, are the tools used for structural variant calling able to handle exome data well (to the extent possible with targeted sequencing?)

The documentation has a small chapter stating that the authors recommend supplying a BED file with the targeted regions, but there is not so much explanation of what the effect of this is.

This is a relevant comment, and we want to clarify that Sarek has been run on whole-exome sequencing data in pilot user projects, and has been reported to us to work well (personal communication), but no comprehensive benchmark has been performed by us to evaluate this. This has been clarified in the Sarek documentation and in the manuscript.
The authors demonstrate that Sarek is both fast and accurate by running it on a tumor/normal (germline) dataset from a previous benchmark study. I think this is acceptable/sufficient, but one could always wish for more; the paper could be strengthened by for example running the well-known public germline HG001 sample against the Genome In a Bottle gold standard dataset.

This is a good suggestion and we have run germline variant calling with Sarek on the HG001 sample and compared to the Genome In a Bottle gold standard dataset, and these results are now included in the manuscript.

Accurate somatic variant calling is difficult. But the included benchmark study demonstrates that Sarek performs well in comparison with other pipelines. The tool leaves it up to the user to decide whether to use output from a single variant caller or the union or intersection from all tools for increased sensitivity or precision.

In summary, I think Sarek is a great addition to the community and recommend the paper for indexing.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

**Competing Interests:** No competing interests were disclosed.
The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com