



Check for updates

SOFTWARE TOOL ARTICLE

eXNVerify: coverage analysis for long and short-read sequencing data in clinical context

[version 1; peer review: 2 approved, 1 approved with reservations]

Sebastian Porębski ¹, Tomasz Stokowy²¹Department of Cybernetics, Nanotechnology and Data Processing, Silesian University of Technology, Gliwice, Poland²Department of Clinical Science, University of Bergen, Bergen, Norway

V1 First published: 13 Jun 2022, 11:645
<https://doi.org/10.12688/f1000research.121775.1>
 Latest published: 13 Jun 2022, 11:645
<https://doi.org/10.12688/f1000research.121775.1>

Abstract




Accurate identification of genetic variants to a large extent is based on the type of experimental technology, quality of the material and coverage of sequencing data obtained. The latter, coverage quality, highly influences variant calling accuracy and final diagnosis. Our motivation was to create a tool that will evaluate genome coverage and accelerate the introduction of long-read sequencing to medical diagnostics and clinical practice. The implementation was guided by the ease of use of the tool by users who are not proficient in using complex software. A Docker container is perfect for this purpose. Using Docker's advantages (flexibility, mobility and ease of use of the proposed tools), we created eXNVerify. This is a tool for inspection of clinical data in the context of pathogenic variants search. The tool calculates clinical depth coverage (CDC) – a measure of coverage which we introduce to evaluate loci with pathogenic germline and somatic variants reported in ClinVar. The tool additionally provides visualization options for user-defined genes of interest. Finally, we present examples of BRCA1, TP53, CFTR application and results of a test conducted in the Extensive Sequence Dataset of Gold-Standard Samples for Benchmarking and Development. eXNVerify improves the diagnostic process of patients related to important genetic diseases and facilitates the assessment of genetic samples by diagnosticians. The use of Docker allows to run an analysis package and does not require any special technical preparation. Detailed examples are included in the GitHub [project](#) documentation and the package can be downloaded directly from [DockerHub](#) using the command: `docker pull porebskis/exnverify:1.0`.


Keywords


long-read technology, whole genome sequencing, single nucleotide variants, sequencing coverage


Open Peer Review

Approval Status   

	1	2	3
version 1			
13 Jun 2022	view	view	view

1. **Zebin Zhang** , Stockholm University,
Stockholm, Sweden

2. **Xueyi Dong** , The Walter and Eliza Hall
Institute of Medical Research, Victoria,
Australia

3. **Wouter De Coster** , University of Antwerp,
Antwerp, Belgium

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the **Bioinformatics** gateway.



This article is included in the **Genomics and Genetics** gateway.

Corresponding author: Tomasz Stokowy (tomasz.stokowy@k2.uib.no)

Author roles: **Porębski S:** Conceptualization, Formal Analysis, Methodology, Software, Visualization, Writing – Original Draft Preparation; **Stokowy T:** Formal Analysis, Funding Acquisition, Investigation, Supervision, Validation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work has been partially supported by the statutory fund no 02/130/BKM21/0010 of the Silesian University of Technology. The Genomics Core Facility at the University of Bergen, which is part of the NorSeq consortium, provided computational support for this study; GCF is supported in part by major grants from the Research Council of Norway (grant number 245979/F50) and Bergen Research Foundation (BFS).

Copyright: © 2022 Porębski S and Stokowy T. This is an open access article 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.

How to cite this article: Porębski S and Stokowy T. **eXNVerify: coverage analysis for long and short-read sequencing data in clinical context [version 1; peer review: 2 approved, 1 approved with reservations]** F1000Research 2022, 11:645 <https://doi.org/10.12688/f1000research.121775.1>

First published: 13 Jun 2022, 11:645 <https://doi.org/10.12688/f1000research.121775.1>

Introduction

Accurate identification of clinically relevant genomic variants strictly depends on sequencing coverage of sequencing data. Long-read (LR) sequencing covers a higher percentage of the human genome than short-read (SR) sequencing and results in more stable coverage¹. Consequently, single nucleotide, indel², and structural variants³ are detected more accurately. Recent benchmarks evaluate the accuracy, precision, and recall of variant calling in long-read genome data⁴; however, the introduction of new findings in the clinical and diagnostic setting requires more time. To accelerate the development of clinical genomics we present eXNVerify (named from “exon and single nucleotide variant verification”), a standalone tool that evaluates and visualizes genome coverage in a clinical context. While the available software approaches can analyze the sequencing data, none of them focuses on evaluating single nucleotide variant (SNV) coverage in the context of diagnostic procedures. This gap is filled by eXNVerify. The comprehensive quality control of medically relevant genes can be now adjusted to the diagnostic procedure. Moreover, the tool helps to verify the sequencing sample in terms of coverage of selected genes or to evaluate the overall genome/exome in terms of variant coverage.

Methods

Operation

eXNVerify consists of two procedures prepared in Python 3.8 with utilization of well-known numerical data-related libraries: `numpy`, `pandas` and `matplotlib`. Hence, for proper utilization of our tool, the user needs to install above packages in their Python distribution on their computer system. Source codes can be executed with the `python` command on Windows or Unix systems. However, we decided to publish a ready-to-go Docker container that includes all dependencies. If the potential user chooses the container, only the Docker application needs to be prepared and then our container can be pulled from the DockerHub repository with one line command: `docker pull porebskis/exnverify:1.0`.

Implementation

The eXNVerify tool is designed to run clinically relevant coverage analysis for SR and LR data, providing integration with the [ClinVar](#) database. The software is designed as two standalone procedures: `geneCoverage` and `snvScore`. The primary input file for both procedures is the coverage record (BED format) obtained from processing BAM files. To create an input file, the user can use dependencies such as `bedtools`⁵, `mosdepth`⁶, or `samtools`⁷ (see our [GitHub documentation](#)).

The first procedure is `geneCoverage`. According to the location of exons of the selected gene, it presents coverage in a graphical form (coverages of exons are light blue fragments in [Figure 1](#), [Figure 2](#) and [Figure 3](#)) with the location of pathogenic SNVs. Germline and somatic variants are shown as red and dark blue dots, respectively. Moreover, `geneCoverage` counts the coverage of these SNVs and summarizes the results in a tabular form. The `geneCoverage` script, in addition to the exon list, pathogenic germline, and pathogenic somatic SNV list, also takes the names of the genes and the coverage threshold as a parameter. The latter is set to evaluate the sample if the gene-related variants are sufficiently covered. Thus, `geneCoverage` reports the percentage of SNV covered for a given gene and includes insufficient coverage in the generated figure. That is, specific exons that are poorly covered

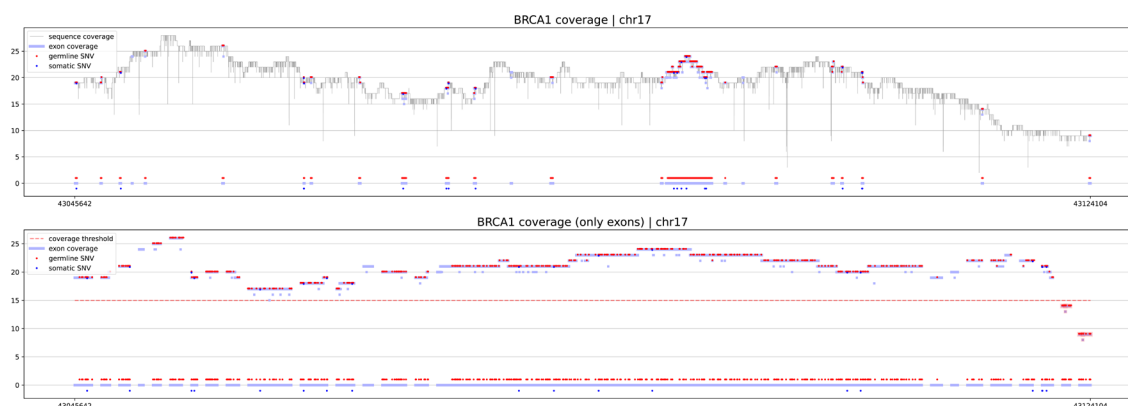


Figure 1. BRCA1 coverage for sample PacBio Long Read. The upper panel demonstrates the distribution of coverage in the region of the gene (exons and introns). The lower panel depicts coverage of exons. X-axis is a genomic locus, specified by the user. Dots highlight positions where pathogenic germline (red) and somatic (dark blue) ClinVar variants are located. If coverage of exons is lower than the threshold specified by the user, they are highlighted by the software in red color.

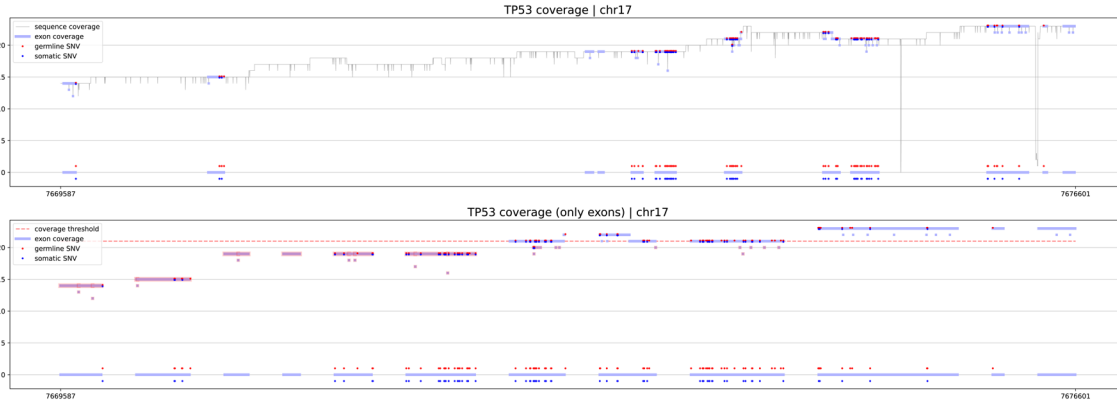


Figure 2. TP53 coverage for PacBio Long Read sample. Coverage of more than 40% of the gene did not reach expected 20x coverage. In such case diagnostic lab should consider optimization of the sequencing protocol, especially in exons 1, 2, 5 and 6, which include germline and somatic pathogenic variants.

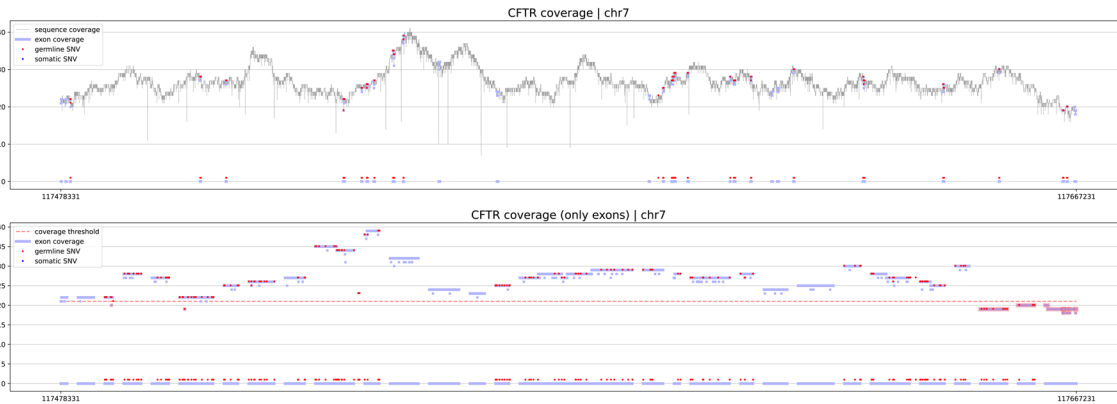


Figure 3. CFTR coverage for PacBio Long Read sample. eXNVerify scales visualization to correctly illustrate genes with large number of exons. In this case, CFTR gene, responsible for development of cystic fibrosis consist of 27 exomes.

(which may contain key variants) are highlighted in red (see Figure 1, Figure 2 and Figure 3). This design helps to evaluate the reliability of the data before and after specific variant calling. Importantly, it is possible to prepare their own reference files with desired exon regions and SNV positions by following the examples provided in the referenced GitHub repository (*Underlying data*⁸).

An additional element of eXNVerify that focuses on the overall evaluation of sequence coverage is snvScore. It is used to check the coverage of all SNVs downloaded from the ClinVar database. The snvScore script checks variant coverage by all chromosomes and provides basic statistics. Finally, snvScore calculates a proposed measure of variant coverage, called clinical depth coverage (CDC), calculated as:

$$CDC^{(t)} = \arg \min_{m \in \mathbb{N}_+} \sum_{i=1}^{N^{(t)}} \text{sgn} \left[C(v_i^{(t)}) - m \right], t = \begin{cases} g & \text{for germline} \\ s & \text{for somatic,} \end{cases} \#(1)$$

where $C(v_i^{(t)})$ is the coverage of the i -th SNV and $N^{(t)}$ is the number of all referenced variants. Germline and somatic variants are analyzed separately; hence t equals g or s for germline and somatic, respectively. We created examples for highly relevant genes for medical genetics and cancer genomics (see Supplementary Files 1 and 2, *Extended data*⁸).

The sample BED file contains exclusive fragments of a gene sequence. Each fragment is related to one value of coverage. An exome reference list is also a BED format file, but it is different since it contains the location of all exons. It means that each row usually expresses a large fragment of sequence. Therefore, one fragment may be covered in a different number of reads. During the implementation, the task was to locate all fragments in the sample BED file that are in one exon fragment from the reference BED. In this way, coverages in one exon

fragment are extracted. Next, if SNV location is available, it is possible to extract coverage information in formerly extracted exon coverage information. Coverage of SNVs is also presented graphically. Moreover, SNV coverage information is aggregated and summarized in table form in a report file. These operations are the core of the geneCoverage procedure.

The second procedure, snvScore explores the whole genome/exome and extracts coverage of all referenced SNVs coverage. Hence, snvScore requires a sample BED file and two ClinVar SNV tables. Exome reference and gene name are not necessary. The sample BED may be a large file, hence snvScore iteratively loads one-chromosome fragments, and extracts and aggregates information about SNV coverage. When finished, it reports CDC (1) for the whole sample with a via-chromosome table of germline and somatic pathogenic SNV coverage statistics. Supplementary File 1 (*Extended data*)⁸ provides a detailed example of snvScore execution.

Results

eXNVerify is a new tool created to evaluate and visualize gene coverage in a clinical context. The tool consists of two methods implemented in Python: geneCoverage and snvScore. The first tool, geneCoverage looks for a gene (or multiple genes) of interest and evaluates it, integrating the coverage with the ClinVar pathogenic variant information. It demonstrates exons in a gene of interest, highlighting positions of pathogenic variants in the ClinVar database (Figure 1, BRCA1 gene). The tool includes both germline pathogenic and somatic pathogenic SNVs. The tool is flexible and suitable for both oncology project (Figure 2, TP53 gene) and rare disease projects (Figure 3, CFTR gene). Processing the samples with the pandas and numpy libraries as well as visualization of the results with the matplotlib library is enough to provide intuitive support for the diagnostician. Moreover, geneCoverage indicates positions in which desired coverage has not been achieved and therefore variant analysis may lead to false negative/positive calls. The tool is suitable for LR and SR data providing novel insights and analysis options for all technologies used currently in clinical laboratories. To address the spectrum of technology-dependent coverage differences we present results for LR whole genome, SR whole genome and SR exome in Supplementary Figures 1 A, B, and C (*Extended data*)⁸; respectively. The second method, snvScore calculates coverage statistics for pathogenic variants, allowing the user to estimate the percentage of all SNVs that are covered above the defined threshold. Table 1 summarizes the essentials of its execution for test samples (Supplementary File 3, *Extended data*)⁸.

Use cases

geneCoverage.py performs detailed verification of pathogenic germline and somatic SNVs for chosen gene(s) in a graphical form. Execution of the procedure require parameters in following order:

```
geneCoverage [-h]
               SampleBED RefExomeBED SNVGermlineTXT SNVSomaticTXT
               Threshold GeneName_s [GeneName_s ...]

positional arguments:
  SampleBED           Path to the mosdepth per-base BED output
  RefExomeBED         Path to the all exons BED file
  SNVGermlineTXT      Path to Clivar-generated table with pathogenic germline SNVs
  SNVSomaticTXT       Path to Clivar-generated table with pathogenic somatic SNVs
  Threshold           Coverage quality threshold
  GeneName_s          Gene name(s)

optional arguments:
  -h, --help          show this help message and exit

Exemplar execution of Docker container eXNVerify with geneCoverage.py on
particular HG003 pacbio-hifi sample for verification of coverage of BRCA1
gene and all somatic and germline SNVs is as follows (code is split to couple
of lines:

docker run -it --rm -v ~/hostpath:/input \
-v ~/hostpath:/output -v ~/hostpath/refs:/refs \
porebskis/exnverify:1.0 ./geneCoverage.py \
input/HG003.pacbio-hifi.21x.haplotag.grch38.bam.per-base.bed \
refs/Exome_Reference_refined.bed refs/SNV_patho_germline.txt \
refs/SNV_patho_somatic.txt \
15 BRCA1
```

Table 1. Results obtained for test samples. CDC: clinical depth coverage; s: somatic; g: germline.

Sample	Sample type	Expected mean coverage	CDC ^(g)	CDC ^(s)	Sample coverage median	% of germline variants covered above threshold	% of somatic variants covered above threshold
HG003	PacBio Long Read	21	20 ± 6	20 ± 5	22	81% (15x)	91% (15x)
HG003	Illumina WGS	20	24 ± 6	24 ± 6	24	94% (15x)	98% (15x)
HG003	Illumina Exome Agilent	100	141 ± 100	162 ± 108	18*	73% (100x)	84% (100x)

The crucial result of abovementioned procedure is graphical file with coverage analysis results of BRCA1 genome sequence (see [Figure 1](#)). More examples and results can be directly downloaded from [8](#).

snvScore.py analyses the whole genome sequence coverage and evaluate all pathogenic germline and somatic SNV coverage quality. Its execution needs to fit the procedure positional arguments as follows:

```
snvScore [-h] SampleBED SNVGermlineTXT SNVSomaticTXT [Threshold]

positional arguments:
  SampleBED           Path to the mosdepth per-base BED output
  SNVGermlineTXT      Path to Clivar-generated table with pathogenic germline SNVs
  SNVSomaticTXT       Path to Clivar-generated table with pathogenic somatic SNVs
  Threshold           SNV coverage quality threshold (optional, positive)

optional arguments:
  -h, --help          show this help message and exit
```

Exemplar snvScore.py execution within eXNVerify Docker container for sample coverage BED file is as follows:

```
docker run -it --rm -v ~/hostpath/:/input -v ~/hostpath/:/output \
-v ~/hostpath/refs/:/refs porebskis/exnverify:1.0./SNVScore.py \
input/HG003.pacbio-hifi.21x.haplotag.grch38.bam.per-base.bed \
refs/SNV_patho_germline.txt refs/SNV_patho_somatic.txt 15
```

The aim of snvScore.py is to prepare coverage analysis of all referenced SNVs in the tabular form ([Table 1](#)). summarizes the results of snvScore execution of different genome sequence data. In the project documentation⁸, the reader may find snvScore results in tabular via-chromosome qualitative results.

Conclusions

The tool can be used to inspect structural variants observed in the sample, especially deletions and copy number changes. This approach can be helpful in a manual verification of structural variants, which is still a recommended practice in medical genetics⁹.

Finally, eXNVerify gives insights into the sample's usefulness in a hypothesis-free analysis of pathogenic variants. The proposed CDC measure provides a percentage of variants covered above the desired threshold in a specified case ([Table 1](#) and Supplementary File 3, *Extended data*,⁸). This measure is useful for everyday laboratory practice to maintain and maximize the quality of experiments. Results of such analyses are provided in [Table 1](#), which indicates the percentage of germline and somatic pathogenic variants specified above the desired threshold. It can also be observed that pathogenic variant coverage differs from median coverage and mean coverage of the sample. For the HG003 PacBio Long Read sample, clinical depth coverage equaled 20x, while global median coverage was 22x. A user of the software can also see that 81% of germline pathogenic variants were covered at least 15x.

We conclude that CDC measures and the percentage of variants covered above the threshold are useful for medical genetics and cancer diagnostics. In summary, our new tool introduces new, easily applicable options for medical genome analysis.

Software availability

Ready-to-go Docker container can be pulled from <https://hub.docker.com/r/porebskis/exnverify>. Source code available from: <https://github.com/porebskis/eXNVerify>. Archived source code as at time of publication: <https://doi.org/10.5281/zenodo.6541899>

License: MIT

Data availability

Underlying data

Test samples were taken from the public repository provided by [Google Cloud Storage](#). The only requirement for users to browse this repository is to have Google account. These data are released under CC-0 license and introduced by Baid *et al.*, 2020⁴. Instructions for accessing this public data can be found in [Google Cloud Storage](#) documentation. For user consideration, we provide the following public links to HG003 samples, generated with three different sequence technologies: [PacBio Long Read](#) (42.1 GB), [Illumina WGS](#) (38.9 GB), and [Illumina Exome Agilent](#) (8.4 GB)

Extended data

GitHub: <https://github.com/porebskis/eXNVerify/tree/main/suppdata> This project contains the following extended data:

S1 Fig BRCA1 coverage for samples: A – PacBio Long Read, B – Illumina WGS, C – Illumina Exome Agilent. Detail description as for Fig 1.

S1 File Supplementary Data – exemplar use cases of eXNVerify with quantitative and graphical results

S2 File geneCoverage report for HG003 PacBio LR, Illumina WGS, Illumina Exome Agilent

S3 File snvScore report for HG003 PacBio LR, Illumina WGS, Illumina Exome Agilent

Acknowledgments

We would like to acknowledge SnotraBio for sharing computational resources which were used to develop the study. We are thankful for constructive insights from employees of the Medical Genetics Department, Haukeland University Hospital, Bergen Norway, especially from Aashish Srivastava, Rita Holdhus and Sigrid Erdal.

References

1. Wenger AM, Peluso P, Rowell WJ, *et al.*: **Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome.** *Nat Biotechnol.* 2019; **37**(10): 1155–1162.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
2. Kolesnikov A, Goel S, Nattestad M, *et al.*: **DeepTrio: Variant Calling in Families Using Deep Learning.** *bioRxiv.* 2021; 2021.04.05.438434.
[Publisher Full Text](#)
3. Mahmoud M, Gobet N, Cruz-Dávalos DI, *et al.*: **Structural variant calling: the long and the short of it.** *Genome Biol.* 2019; **20**(1): 246.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Baid G, Nattestad M, Kolesnikov A, *et al.*: **An Extensive Sequence Dataset of Gold-Standard Samples for Benchmarking and Development.** *bioRxiv.* 2020; 2020.12.11.422022.
[Publisher Full Text](#)
5. Quinlan AR: **Bedtools: the swiss-army tool for genome feature analysis.** *Curr Protoc Bioinformatics.* 2014; **47**: 11.12.1–34.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
6. Pedersen BS, Quinlan AR: **Mosdepth: quick coverage calculation for genomes and exomes.** *Bioinformatics.* 2018; **34**(5): 867–868.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
7. Li H, Handsaker B, Wysoker A, *et al.*: **The sequence alignment/map format and samtools.** *Bioinformatics.* 2009; **25**(16): 2078–2079.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
8. Porebski S: **porebskis/eXNVerify, Exon and SNV coverage verification, software, v1.0.1.** 2022.
<http://www.doi.org/10.5281/zenodo.6541899>
9. Minoche AE, Lundie B, Peters GB, *et al.*: **ClinSV: clinical grade structural and copy number variant detection from whole genome sequencing data.** *Genome Med.* 2021; **13**(1): 32.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status:   

Version 1

Reviewer Report 24 January 2024

<https://doi.org/10.5256/f1000research.133674.r228027>

© 2024 De Coster W. 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.



Wouter De Coster 

University of Antwerp, Antwerp, Belgium

The authors describe eXNVerify, a tool to assess the coverage in sequencing data. This is relevant, as coverage is a crucial parameter for variant detection and accuracy. The tool is well developed and the plots are mostly clear, but overall the scope of this tool is relatively narrow - focusing on one sample at a time. Please find my detailed comments below.

In the introduction, the authors mention that long-read sequencing covers a higher percentage of the human genome and results in more stable coverage. While the stable coverage is probably mostly a feature of the absence of PCR amplification, I would mainly warrant some caution about the sentence following that statement: 'Consequently, single nucleotide, indel, and structural variants are detected more accurately'. I believe this statement to be incorrect, specifically in the case of short indels - as such variants are the dominant error mode of nanopore sequencing. While long reads enable alignment in highly repetitive regions, and thus variant calling there, some more nuance is warranted with regard to which variants are more accurately detected. I am a bit unsure if the focus on long-read sequencing in the introduction is relevant, as your tool appears to be broadly applicable.

I am a bit confused by the bottom panels of Figures 1 and 3 and both panels in Figure 2. Do these plots contain two samples, with the blue lines of exon coverage at $y=0$? Why are variants shown twice, once in the plot at different heights and once at $y=1/-1$?

The legend of Figure 3 mentions that it consists of "27 exomes". which should presumably be 27 exons.

Can the authors elaborate on what the coverage threshold of 15x was chosen for Figure 1, and 20x for Figure 2? I also think that the converge threshold line in Figure 2 is not at 20x but just above that.

I think distributing your tool via docker might make it easier for some people, as installation becomes easier, but then requires that docker is already installed, which may not be the case. I would suggest that you include installation instructions (using pip or conda) for those users that

cannot use docker. Regrettably, not all sysadmins will allow users to use docker.

I see in the GitHub documentation that the tool requires a 'Path to Clivar-generated table with pathogenic germline SNVs,' but it is not clear how that one needs to be generated. Example files are provided in the repository, which is highly appreciated, but it is unclear to which reference genome they correspond. It would however be most useful if your tool included a command to download such a table automatically from the ClinVar database. The same argument could be made for the exome bed file. In addition, the authors recommend Mosdepth, which is an excellent choice, but I would also suggest including optimal parameters for this tool in your documentation, as it has various options. I see the SNVSomaticTXT is a required argument, but I can imagine that somatic variants are not necessarily relevant for everyone. Maybe that could be an optional input, and maybe a suitable default could already be set for the threshold. In addition, if users have many genes of interest, they will have to provide them all one by one on the command line, which is probably frustrating.

The example docker command for geneCoveray.py in the GitHub documentation misses a trailing slash '\', and thus yields a confusing error. It also appears the coverage bed file cannot be (b)gzip compressed? That is very unfortunate and something I would suggest adding.

While not critical, I think it would be interesting if users could provide multiple samples as input (e.g. a trio), for comparison, rather than having to run the tool multiple times and open multiple separate plots.

That said, after downloading all files, the tool ran as expected and the plots provided a useful representation of the coverage.

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?

Partly

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

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: bioinformatics, data visualization, sequencing data analysis

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, however I have

significant reservations, as outlined above.

Reviewer Report 02 January 2024

<https://doi.org/10.5256/f1000research.133674.r228033>

© 2024 Dong X. 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.



Xueyi Dong 

The Walter and Eliza Hall Institute of Medical Research, Victoria, Australia

In this article, the authors introduced their software tool eXNVerify, which provides functions to visualize and check variant coverage of genes.

Overall, the functions and usage of eXNVerify are introduced clearly in this article. The software tool shows good practical value in making it easier to investigate SNV coverage from sequencing data. The format of visualization that eXNVerify takes is also clear and helpful. However, I have several minor comments:

1. In the exon-only coverage plots (Figure 1-3 bottom panels), the x-axis is scaled to skip the intron regions and highlight the exon regions. It helps show more details of SNVs in exon regions, especially for genes with lots of exons. However, the starting and ending point labels on the x-axis are misleading. It should be reflected on the plots of how the axis was scaled, such as including a scale bar of genomic positions.
2. The software tool is provided as a docker image, which simplifies the installation procedure. However, some institution computers and/or servers don't support docker. Are alternative installation methods available?

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**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.**

Reviewer Report 27 June 2022

<https://doi.org/10.5256/f1000research.133674.r140681>

© 2022 Zhang Z. 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.

**Zebin Zhang** 

Division of Population Genetics, Department of Zoology, Stockholm University, Stockholm, Sweden

Porębski S and Stokowy T provide a software, "eXNVerify," which evaluates and visualizes genome and variations coverage for long and short reads sequencing data in the clinical context. This software is useful, especially for users who are not proficient in bioinformatics, as it is easy to install, use, and visualize results. It also filled the gap in evaluating the coverage of SNV in the context of diagnostic procedures.

While much of the descriptions and results are clear, there are some minor concerns about this paper. These concerns are laid out below:

1. The authors declare that their software can work on long and short reads sequencing data in the title. I believe this as the authors exhibited long-read results in the results section and short-read results in the supplementary section. However, in the abstract, the authors describe, "Our motivation was to create a tool that will evaluate genome coverage and accelerate the introduction of long-read sequencing to medical diagnostics and clinical practice," without mentioning short-read sequencing. Is this implying that the result for the short read is not as reliable as the long read?
2. Methods - operation section. The source codes seem only work on Windows and Linux systems. Because many people work on this system, especially those who are not proficient in bioinformatics or can't access any servers. So my question is, does the eXNVerify work on the macOS system? If not, please explain why.
3. The function of highlighting the insufficient coverage is pretty good, and the authors give some examples of the coverage thresholds 15x and 20x. Why did the authors choose those two values? Are these values arbitrary or selected based on some statistical consideration? Can the software give users a recommended value for the threshold?

4. In fig 3 and 4, the threshold line of 20 doesn't match the y-axis.
5. Given the importance of de novo mutation in clinical, is that possible for authors to add another function of output the coverage of those variations?
6. Mapping quality is also critical in variants calling, do the authors ever consider quality in variants evaluation and CDC calculation?

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

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.

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

F1000Research