

covtobed: a simple and fast tool to extract coverage tracks from BAM files

Giovanni Birolo¹ and Andrea Telatin²

¹ Dept. Medical Sciences, University of Turin, ITALY ² Gut Microbes and Health Programme, Quadram Institute Bioscience, Norwich, UK

DOI: [10.21105/joss.02119](https://doi.org/10.21105/joss.02119)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [William Rowe](#) ↗

Reviewers:

- [@jdeligt](#)
- [@brentp](#)

Submitted: 30 January 2020

Published: 06 March 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

A common task in bioinformatics is the mapping of DNA sequencing reads (produced by “next generation sequencing” experiments) against a reference genome. The output of the alignment is commonly encoded in a BAM file (Li et al., 2009). For several applications of DNA sequencing it is useful to extract the *depth of coverage* (Sims, Sudbery, Ilott, Heger, & Ponting, 2014) at specific positions in the BAM file, encoding the output in the standard BED format (Quinlan & Hall, 2010).

Here we describe *covtobed*, a C++ program designed to extract the depth of coverage per position from a sorted BAM file, optionally specifying a range of coverage of interest and a minimum length for the features to be printed in the output BED file. Parsing of BAM files is performed using *libbamtools* (Barnett, Garrison, Quinlan, Strömberg, & Marth, 2011).

The design has been inspired by the UNIX programming philosophy (Wikipedia contributors, 2019), and thus *covtobed* performs a single task and supports input and output streams.

Availability and Installation

covtobed is distributed with MIT licence and available from the [GitHub repository](#), and can be easily installed via Miniconda from the “bioconda” channel (*i. e.* `conda install -c bioconda covtobed`).

The tool is also available as a Docker image downloadable from [Docker Hub](#) (*i. e.* `docker pull andreatelatin/covtobed`) or as a Singularity image.

Code (structure and dependencies)

The code is object oriented, including an *Input* class handling reading, parsing and filtering of alignments and an *Output* class handling coverage filtering and writing in different formats. The main algorithm is based on a *priority_queue* from the standard library and is both fast and memory efficient.

covtobed relies on *libbamtools* (Barnett et al., 2011) for BAM file parsing, and *cpp-optparse* (Weißl, 2017) for command line option parsing.

Documentation

The package documentation is maintained in the [GitHub wiki](#). The documentation contains examples of usage, example of produced output and details about the package.

Example applications

When performing target enrichment experiments (where the aim is to sequence a set of selected regions of a genome), it's important to detect a lack of coverage or insufficient coverage (*i.e.* the coverage on target is lower than THRESHOLD). This information can be obtained by intersecting (using *bedtools*, (Quinlan & Hall, 2010)) the BED file describing the captured target regions (usually supplied by the company producing the kit) with the output of *covtobed*.

The tool has been used, for example, in the setup of a *target enrichment* panel targeting 71 human genes (Poloni et al., 2019), in order to detect uncovered regions.

While a tool exists – called *mosdepth* (Pedersen & Quinlan, 2018) – to perform a coverage analysis, *covtobed* was designed with the ability to quickly extract regions between user-defined coverage intervals and, more importantly, with streaming from standard input and to standard output, that *Mosdepth* doesn't support. *covtobed* is available both for Linux and macOS, while *mosdepth* is only available for Linux, and this makes *covtobed* a suitable building block for diverse pipelines (*e. g.* microbial genomics requires lesser resources and it is not uncommon to perform complete analyses on a laptop).

Performance

covtobed is a fast tool, constantly outperforming the popular *bedtools* and providing comparable speed with *mosdepth*. With some datasets, like “gene panels”, *covtobed* is more than ten times faster than *mosdepth*.

The scripts to perform the benchmark are available in the [github repository](#).

Acknowledgements

The authors gratefully acknowledge the support of the Biotechnology and Biological Sciences Research Council (BBSRC); this research was partly supported by the BBSRC Institute Strategic Programme Gut Microbes and Health BB/R012490/1 and its constituent project BBS/E/F/000PR10353. Analyses and benchmark performed using the MRC CLIMB cloud computing environment supported by grant MR/L015080/1.

References

- Barnett, D. W., Garrison, E. K., Quinlan, A. R., Strömberg, M. P., & Marth, G. T. (2011). BamTools: A c++ api and toolkit for analyzing and managing bam files. *Bioinformatics (Oxford, England)*, 27(12), 1691–1692. doi:[10.1093/bioinformatics/btr174](https://doi.org/10.1093/bioinformatics/btr174)
- Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., et al. (2009). The sequence alignment/map format and samtools. *Bioinformatics (Oxford, England)*, 25(16), 2078–2079. doi:[10.1093/bioinformatics/btp352](https://doi.org/10.1093/bioinformatics/btp352)

- Pedersen, B. S., & Quinlan, A. R. (2018). Mosdepth: Quick coverage calculation for genomes and exomes. *Bioinformatics (Oxford, England)*, *34*(5), 867—868. doi:[10.1093/bioinformatics/btx699](https://doi.org/10.1093/bioinformatics/btx699)
- Poloni, G., Calore, M., Rigato, I., Marras, E., Minervini, G., Mazzotti, E., Lorenzon, A., et al. (2019). A targeted next-generation gene panel reveals a novel heterozygous nonsense variant in the tp63 gene in patients with arrhythmogenic cardiomyopathy. *Heart rhythm*, *16*(5), 773—780. doi:[10.1016/j.hrthm.2018.11.015](https://doi.org/10.1016/j.hrthm.2018.11.015)
- Quinlan, A. R., & Hall, I. M. (2010). BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics*, *26*(6), 841–842. doi:[10.1093/bioinformatics/btq033](https://doi.org/10.1093/bioinformatics/btq033)
- Sims, D., Sudbery, I., Illott, N. E., Heger, A., & Ponting, C. P. (2014). Sequencing depth and coverage: Key considerations in genomic analyses. *Nature reviews. Genetics*, *15*(2), 121—132. doi:[10.1038/nrg3642](https://doi.org/10.1038/nrg3642)
- WeiBl, J. (2017). Cpp-optparse. *GitHub repository*. <https://github.com/weisslj/cpp-optparse>; GitHub.
- Wikipedia contributors. (2019). Unix philosophy — Wikipedia, the free encyclopedia. Retrieved from https://en.wikipedia.org/w/index.php?title=Unix_philosophy&oldid=919880773