

alignparse: A Python package for parsing complex features from high-throughput long-read sequencing

Katharine H.D. Crawford^{1, 2} and Jesse D. Bloom^{1, 3}

1 Basic Sciences and Computational Biology, Fred Hutchinson Cancer Research Center, Seattle, Washington, USA **2** Department of Genome Sciences and Medical Scientist Training Program, University of Washington, Seattle, Washington, USA **3** Howard Hughes Medical Institute, Seattle, Washington, USA

DOI: [10.21105/joss.01915](https://doi.org/10.21105/joss.01915)

Software

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

Editor: [William Rowe](#) ↗

Reviewers:

- [@bede](#)
- [@afrubin](#)

Submitted: 19 November 2019

Published: 11 December 2019

License

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

Summary & Purpose

Advances in sequencing technology have made it possible to generate large numbers of long, high-accuracy sequencing reads. For instance, the new PacBio Sequel platform can generate hundreds of thousands of high-quality circular consensus sequences in a single run (Hebert et al., 2018; Rhoads & Au, 2015). Good programs exist for aligning these reads for genome assembly (Chaisson & Tesler, 2012; Li, 2018). However, these long reads can also be used for other purposes, such as sequencing PCR amplicons that contain various features of interest. For instance, PacBio circular consensus sequences have been used to identify the mutations in influenza viruses in single cells (Russell, Elshina, Kowalsky, Velthuis, & Bloom, 2019), or to link barcodes to gene mutants in deep mutational scanning (Matreyek et al., 2018). For such applications, the alignment of the sequences to the targets may be fairly trivial, but it is not trivial to then parse specific features of interest (such as mutations, unique molecular identifiers, cell barcodes, and flanking sequences) from these alignments.

Here we describe [alignparse](#), a Python package for parsing complex sets of features from long sequences that map to known targets. Specifically, it allows the user to provide complex target sequences in Genbank Flat File format that contain an arbitrary number of user-defined sub-sequence features (Sayers et al., 2019). It then aligns the sequencing reads to these targets and filters alignments based on whether the user-specified features are present with the desired identities (which can be set to different thresholds for different features). Finally, it parses out the sequences, mutations, and/or accuracy (sequence quality) of these features as specified by the user. The flexibility of this package therefore fulfills the need for a tool to extract and analyze complex sets of features in large numbers of long sequencing reads.

Uses & Examples

Below are two example use cases of [alignparse](#) from our research. Code, data, and example output are included in the [alignparse](#) documentation.

Sequencing deep mutational scanning libraries

In deep mutational scanning experiments, researchers use mutant libraries to assay the effects of tens of thousands of individual mutations to a gene-of-interest in one experiment (Fowler & Fields, 2014). One way to make deep mutational scanning of long gene variants work efficiently with short-read Illumina sequencing is to link the mutations in each variant to a

unique molecular barcode (Hiatt, Patwardhan, Turner, Lee, & Shendure, 2010). This barcode linking can be done by long-read PacBio sequencing of the variant library (Matreyek et al., 2018), but it is then necessary to parse the resulting long reads to associate the barcode with the mutations in the variant.

The [alignparse](#) package provides a standard tool for parsing barcodes and linked mutations from the long-read sequencing data. It also allows for the parsing of additional sequence features necessary for validating the quality of deep mutational scanning libraries, such as the presence of terminal sequences or other identifying tags. The [RecA deep mutational scanning library example](#) demonstrates this use.

Single-cell viral sequencing

Some viral genomes are sufficiently small to be sequenced in their entirety using long-read sequencing technology. Recent work has shown that such long-read sequencing of viral genes can be combined with standard single-cell transcriptomic technologies (such as 10x Chromium) to simultaneously sequence the infecting virus and characterize the transcriptome in single infected cells (Russell et al., 2019). Such experiments require parsing the long-read viral sequences to identify viral mutations as well as cell barcodes, unique molecular identifiers, and other flanking sequences. The [single-cell virus sequencing example](#) shows how such parsing can readily be performed using [alignparse](#).

How alignparse works

[alignparse](#) takes the following inputs:

1. One or more user-defined Genbank files containing the sequence of one or more alignment targets with an arbitrary number of user-defined features. These Genbank files can be readily generated using sequence editing programs, such as [ApE](#) or [Benchling](#).
2. A YAML file containing parsing specifications for each feature. These specifications include filters indicating the maximal allowed mutations in each feature, as well as information on what output should be parsed for each feature (e.g., its sequence, its mutations, or simply if it is present).
3. A FASTQ file containing the long-read sequencing data. This file can be gzipped. There is no need to decompress gzipped FASTQ files first.

These inputs are used to define a [Targets](#) object. [alignparse](#) then uses this [Targets](#) object to create sequence alignments and parse sequence features defined in the input Genbank and YAML files.

[alignparse](#) aligns sequencing reads to the targets using [minimap2](#). The [alignparse.minimap2](#) submodule provides alignment specifications optimized for the two example use cases described above. [alignparse](#) uses the [cs tags](#) generated by [minimap2](#) to extract the relevant features from the alignments into intuitive data frames or CSV files.

We expect most users to align sequences and parse features in a single step using the [alignparse.targets.Targets.align_and_parse](#) function. However, these aligning and parsing steps can be carried out separately, as seen in the [Lassa virus glycoprotein](#) example. Indeed, the [alignparse.targets.Targets.parse_alignment](#) function should be able to parse features from any alignment file (in [SAM format](#)) as long as the alignments have [cs tags](#) and a corresponding [Targets](#) object has been defined that identifies the targets to which the query sequences were aligned and specifies the features to parse and filters to use.

Downstream analyses of parsed features are facilitated by the [alignparse.consensus](#) submodule. This submodule provides tools for grouping reads by shared barcodes, determining consensus sequences for barcoded reads, and further processing mutation information for downstream analyses. Since the main outputs from [alignparse](#) are in intuitive data frame formats, downstream analyses can be highly customized by the user. Thus, [alignparse](#) provides a flexible and useful tool for parsing complex sets of features from high-throughput long-read sequencing of pre-defined targets.

Code Availability

The [alignparse](#) source code is on GitHub at <https://github.com/jbloomlab/alignparse> and the documentation is at <https://jbloomlab.github.io/alignparse>.

Acknowledgements

We would like to thank members of the Bloom lab for helpful discussions and beta testing. This work was supported by the following grants from NIAID of the NIH: R01 AI141707 and R01 AI140891. JDB is an Investigator of the Howard Hughes Medical Institute.

References

- Chaisson, M. J., & Tesler, G. (2012). Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): Application and theory. *BMC Bioinformatics*, *13*(238). doi:[10.1186/1471-2105-13-238](https://doi.org/10.1186/1471-2105-13-238)
- Fowler, D. M., & Fields, S. (2014). Deep mutational scanning: A new style of protein science. *Nature Methods*, *11*(8). doi:[10.1038/nmeth.3027](https://doi.org/10.1038/nmeth.3027)
- Hebert, P. D. N., Braukmann, T. W. A., Prosser, S. W. J., Ratnasingham, S., deWaard, J. R., Ivanova, N. V., Janzen, D. H., et al. (2018). A sequel to sanger: Amplicon sequencing that scales. *BMC Genomics*, *19*(219). doi:[10.1186/s12864-018-4611-3](https://doi.org/10.1186/s12864-018-4611-3)
- Hiatt, J. B., Patwardhan, R. P., Turner, E. H., Lee, C., & Shendure, J. (2010). Parallel, tag-directed assembly of locally derived short sequence reads. *Nature Methods*, *7*(2). doi:[10.1038/nmeth.1416](https://doi.org/10.1038/nmeth.1416)
- Li, H. (2018). Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics*, *34*(18), 3094–3100. doi:[10.1093/bioinformatics/bty191](https://doi.org/10.1093/bioinformatics/bty191)
- Matreyek, K. A., Starita, L. M., Stephany, J. J., Martin, B., Chiasson, M. A., Gray, V. E., Kircher, M., et al. (2018). Multiplex assessment of protein variant abundance by massively parallel sequencing. *Nature Genetics*, *50*, 874–882. doi:[10.1038/s41588-018-0122-z](https://doi.org/10.1038/s41588-018-0122-z)
- Rhoads, A., & Au, K. F. (2015). PacBio sequencing and its applications. *Genomics, Proteomics & Bioinformatics*, *13*(5), 278–279. doi:[10.1016/j.gpb.2015.08.002](https://doi.org/10.1016/j.gpb.2015.08.002)
- Russell, A. B., Elshina, E., Kowalsky, J. R., Velthuis, A. J. W. te, & Bloom, J. D. (2019). Single-cell virus sequencing of influenza infections that trigger innate immunity. *Journal of Virology*, *93*(14), e00500–19. doi:[10.1128/JVI.00500-19](https://doi.org/10.1128/JVI.00500-19)
- Sayers, E. W., Cavanaugh, M., Clark, K., Ostell, J., Pruitt, K. D., & Karsch-Mizrachi, I. (2019). GenBank. *Nucleic Acids Research*. doi:[10.1093/nar/gkz956](https://doi.org/10.1093/nar/gkz956)