The RAPI Read Aligner API

Luca Pireddu, Gianluigi Zanetti
CRS4, Pula, Italy
Email: luca.pireddu@crs4.it

Web site URL: `https://github.com/crs4/rapi.git`
License: MIT

In work related to NGS data, read aligners are one of the most fundamental pieces of analysis software. In fact, the extensive attention dedicated to the problem has resulted in a number of effective alignment programs, many of which are continuously updated with improvements to alignment quality and efficiency. Aligners are typically packaged as command-line programs that expect to receive three main arguments: the path to the indexed reference sequence, the path to the input read file in Fastq format, and the path to the output file in SAM or BAM format. This simple interface makes a number of assumptions: that all the required data files (reference, input and output) are accessible on locally mounted file systems; that the data are in the formats that the aligner supports; and that the use case supports executing a new process each time an alignment is required. Although these assumptions may appear to be reasonable, they are actually extremely limiting to work aimed at exploring novel programmatic ways to *use* aligners – e.g., interactive analysis, scripting and, more ambitiously, research to advance the standard sequencing pipeline's data flow architecture.

A more flexible approach would be to have read aligners define an API and package their functionality as a library, potentially in addition to the standard command line interface. We propose to the NGS community a general solution: a standard read aligner application programming interface, that can be implemented with any underlying read mapping technology: the *Read Aligner Application Programming Interface (RAPI)*. For maximum compatibility the RAPI is defined in C. It includes generic functions and data structures to support typical alignment operations: index a reference sequence, load and unload the reference, map reads to the reference, interpret the results. Further, RAPI includes Python bindings, making it simple to load an aligner implementing the interface as a Python module and using it in a script or an interactive Python session. The project also includes a reference aligner interface implementation that wraps the BWA-MEM [1] aligner. This aligner plug-in is used in the Seal project [2] to compute sequence alignments on the Hadoop distributed computing platform. The fact that RAPI does not make any assumptions about the source or destination of the data makes it possible to easily integrate scalable computing and data storage technology developed for the Big Data computing problems, such as the Hadoop Distributed File System, into an alignment workflow. Since RAPI also does not make any assumptions about the data formats, it also facilitates using advanced column-oriented file formats, such as AvroParquet, that offer many advantages over conventional file formats in terms of storage efficiency and the ability to selectively read columns of a data set. Finally, since RAPI standardizes the aligner interface, one could parametrize the aligner to use within a RAPI-based pipeline, swapping aligner without changing any of the code.

In summary, we propose to the NGS community a standard aligner application programming interface. The interface defines function calls for alignment operations while remaining agnostic to the file system, data format, and any specifics internal to the aligner. The standard interface makes it simple and safe to harness aligner plug-ins to prototype and create novel functionality. Of course, the interface is released under an open source license and we welcome feedback and contributions from the community.

References

[1] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv*, 2013.

[2] Luca Pireddu, Simone Leo, and Gianluigi Zanetti. Seal: a distributed short read mapping and duplicate removal tool. *Bioinformatics*, 27(15):2159–2160, 2011.