The Journal of Open Source Software

khmer release v2.1: software for biological sequence analysis

Daniel Standage[1], **Ali Aliyari**[2], **Lisa J. Cohen**[1], **Michael R. Crusoe**[4], **Tim Head**[5], **Luiz Irber**[1], **Shannon EK Joslin**[2], **N. B. Kingsley**[2], **Kevin D. Murray**[7], **Russell Neches**[8], **Camille Scott**[1], **Ryan Shean**[9], **Sascha Steinbiss**[10], **Cait Sydney**[11], **and C. Titus Brown**[1]

1 Lab for Data Intensive Biology; School of Veterinary Medicine; University of California, Davis **2** Integrative Genetics and Genomics Graduate Group; University of California, Davis **3** Molecular, Cellular, and Integrative Physiology Graduate Group; University of California, Davis **4** Common Workflow Language Project **5** Wild Tree Tech **6** Computer Science Graduate Group; University of California, Davis **7** ARC Centre of Excellence in Plant Energy Biology; Australian National University **8** Microbiology Graduate Group; Univerity of California, Davis **9** University of Washington **10** Debian Project **11** Google, Inc. **12** Department of Population Health and Reproduction; School of Veterinary Medicine; University of California, Davis

DOI: 10.21105/joss.00272

Software

- Review
- Repository
- Archive

Summary

The khmer software is a set of command-line tools built around a Python library designed for analysis of large DNA sequence collections. Functionality in khmer has primarily been motivated by scaling issues with (meta)genome and (meta)transcriptome assembly (Crusoe et al. 2015). khmer provides convenient access to several k-mer based operations on DNA sequence collections, such as abundance filtering, error trimming, assembly graph partitioning, and most notably, abundance normalization of reads (C. T. Brown et al. 2012) and streaming error trimming of reads (Zhang, Awad, and Brown 2015). All of these operations utilize khmer's implementation of two primary data structures, the Bloom filter and the Count-Min Sketch, for efficient probabalistic storage of k-mer presence or k-mer abundance, respectively (J. Pell et al. 2012; J. A. C.-K. Zhang Qingpeng AND Pell 2014).

Release version 2.1 of the khmer software includes several new features that extend its utility to a wider set of sequence processing and analysis problems. These include the following: support for variable-coverage trimming of sequence reads; support for $k > 32$ using the non-reversible hash function MurmurHash3 (https://github.com/aappleby/smhasher); a new optional Count-Min Sketch implementation providing increased storage efficiency; support for assembly directly from a k-mer graph; a script for computing a compact de Bruijn graph from a k-mer graph; and several examples of khmer's Python and C++ APIs for those interested in using and extending the library.

References

Brown, C. Titus, Adina Howe, Qingpeng Zhang, Alexis B. Pyrkosz, and Timothy H. Brom. 2012. "A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data."

Crusoe, MR, HF Alameldin, S Awad, E Boucher, A Caldwell, R Cartwright, A Charbonneau, et al. 2015. "The Khmer Software Package: Enabling Efficient Nucleotide Sequence

Analysis." *F1000Research* 4 (900). doi:10.12688/f1000research.6924.1.

Pell, Jason, Arend Hintze, Rosangela Canino-Koning, Adina Howe, James M. Tiedje, and C. Titus Brown. 2012. "Scaling Metagenome Sequence Assembly with Probabilistic de Bruijn Graphs." *Proceedings of the National Academy of Sciences* 109 (33): 13272–7. doi:10.1073/pnas.1121464109.

Zhang, Jason AND Canino-Koning, Qingpeng AND Pell. 2014. "These Are Not the K-Mers You Are Looking for: Efficient Online K-Mer Counting Using a Probabilistic Data Structure." *PLOS ONE* 9 (7). Public Library of Science: 1–13. doi:10.1371/journal.pone.0101271.

Zhang, Qingpeng, Sherine Awad, and C. Titus Brown. 2015. "Crossing the Streams: A Framework for Streaming Analysis of Short Dna Sequencing Reads." *PeerJ PrePrints* 3 (March): e890v1. doi:10.7287/peerj.preprints.890v1.