Bioinformatics, 36(9), 2020, 2929–2931
doi: 10.1093/bioinformatics/btaa028
Advance Access Publication Date: 13 January 2020
Applications Note

Gene expression

RiboFlow, RiboR and RiboPy: an ecosystem for analyzing ribosome profiling data at read length resolution

Hakan Ozadam, Michael Geng and Can Cenik*

Department of Molecular Biosciences, University of Texas at Austin, Austin, TX 78705, USA

*To whom correspondence should be addressed.

Associate Editor: Anthony Mathelier

Received on November 25, 2019; revised on January 6, 2020; editorial decision on January 7, 2020; accepted on January 8, 2020

Abstract

Summary: Ribosome occupancy measurements enable protein abundance estimation and infer mechanisms of translation. Recent studies have revealed that sequence read lengths in ribosome profiling data are highly variable and carry critical information. Consequently, data analyses require the computation and storage of multiple metrics for a wide range of ribosome footprint lengths. We developed a software ecosystem including a new efficient binary file format named 'ribo'. Ribo files store all essential data grouped by ribosome footprint lengths. Users can assemble ribo files using our RiboFlow pipeline that processes raw ribosomal profiling sequencing data. RiboFlow is highly portable and customizable across a large number of computational environments with built-in capabilities for parallelization. We also developed interfaces for writing and reading ribo files in the R (RiboR) and Python (RiboPy) environments. Using RiboR and RiboPy, users can efficiently access ribosome profiling quality control metrics, generate essential plots and carry out analyses. Altogether, these components create a software ecosystem for researchers to study translation through ribosome profiling.

Availability and implementation: For a quickstart, please see https://ribosomeprofiling.github.io. Source code, installation instructions and links to documentation are available on GitHub: https://github.com/ribosomeprofiling.

Contact: ccenik@austin.utexas.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Ribosome profiling is a powerful method for measuring transcriptome-wide translation through the sequencing of ribosome-protected mRNA fragments (Ingolia *et al.*, 2009, 2011). This transformative approach has been applied to many organisms and can approximate translational efficiency, a major determinant of protein abundance (Kristensen *et al.*, 2013; Schwanhäusser *et al.*, 2011). Consequently, ribosome profiling studies fulfill a critical gap in our understanding of protein abundance, which is only partially explained by RNA expression (Ly *et al.*, 2014; Marguerat *et al.*, 2012). Moreover, ribosome profiling has proved invaluable in studying the mechanisms of translation (Ingolia *et al.*, 2019).

Initial ribosome profiling studies focused on the classical ~28 nt ribosome-protected footprints (RPFs) corresponding to ribosomes with occupied A-sites. In contrast, recent work revealed that RPF lengths are variable and carry critical information (Lareau *et al.*, 2014; Liakath-Ali *et al.*, 2018; Wu *et al.*, 2019). For example, ribosomes can protect short footprints (15–21 nts), characteristic of different ribosome conformations (Guydosh and Green, 2017; Lareau *et al.*, 2014; Wu *et al.*, 2019) as well as longer footprints (~60 nt) indicative of ribosome collisions (Arpat *et al.* 2019; Guydosh and Green, 2014). Importantly, these key observations signal a new chapter of translation studies that will need to be analyzed by taking variable RPF lengths into account.

Analyses for a wide range of RPF lengths (~15–60 nts) entail increased computational complexity and organizational challenges that were negligible in ribosome profiling studies that focused solely on ~28 nt RPFs. In particular, the naive use of text files would be highly inefficient in organization, computation and storage. Similarly, directly accessing the unprocessed alignment files for analyses limits data portability and efficiency. While a range of computational approaches have been developed for ribosome profiling, they either directly rely on alignment files for analyses or do not consider RPF length as a major design feature (Berg *et al.*, 2019; Birkeland *et al.*, 2018; Carja *et al.*, 2017; Chung *et al.*, 2015; Perkins *et al.*, 2019; Popa *et al.*, 2016). Moreover, existing software often lags behind high-quality standards in installation, ease of use, documentation and portability (Wang *et al.*, 2019).

We introduce a new binary file format called 'ribo' to enable efficient organization of ribosome profiling data including studies focusing on a broad range of RPF lengths. Similar binary formats are widely used to store many sequencing types such as 'bam' for sequence alignments (Li *et al.*, 2009), 'BUS' for single-cell RNA-Seq (Melsted *et al.*, 2019) and 'cooler'/'hic' for Hi-C (Abdennur and Mirny, 2019; Durand *et al.*, 2016). Our new ribo file is designed to

© The Author(s) 2020. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

work with ribosome profiling data at nucleotide length resolution. Ribo files organize all quantification tables and metadata for efficient data storage and retrieval.

Importantly, we designed a software ecosystem around this file format that includes three major components. RiboFlow is a Nextflow (Di Tommaso *et al.*, 2017) based alignment pipeline that generates ribo files from raw sequencing data. RiboFlow can run on local servers, major job schedulers and cloud-based platforms with minimal configuration. We provide a Docker container image to enable deployment of RiboFlow on all major operating systems. Finally, we developed two interfaces, RiboPy and RiboR, to work with ribo files. RiboPy, a Python package, can be used to create ribo files and subsequently analyze and visualize data. RiboR is a package that enables seamless analyses with ribo files in the R environment. Importantly, we offer a superior user experience facilitating data portability, enabling installation via package management tools and providing detailed documentation. Taken as a whole, this ecosystem is a useful platform for researchers to study translation using ribosome profiling.

2 Implementation and availability

2.1 RiboFlow

RiboFlow is a Nextflow (Di Tommaso *et al.*, 2017) based pipeline that generates ribo files from sequencing files in one step. The RiboFlow pipeline begins with adapter removal from raw sequencing reads in fastq format. The clipped reads are then filtered to remove common non-coding RNAs such as ribosomal RNA. Next, remaining reads are mapped to the transcriptome, and alignments with a mapping quality higher than a predetermined value are retained. An optional polymerase chain reaction (PCR) deduplication step can collapse PCR duplicates defined by having the same read length and identical mapping position. Finally, results from multiple experiments are compiled into one ribo file (Fig. 1A).

Using the provided Docker container image, RiboFlow can run on all major operating systems. The number of simultaneous threads can easily be set, allowing for efficient hardware utilization across a wide spectrum of computational resources. RiboFlow can be configured to run on mainstream job schedulers such as LSF, SGE and SLURM as well as cloud environments such as Amazon Web Services and Google Cloud Platform.

To increase the quality and accuracy of the sequence alignments and simplify quantifications, we designed RiboFlow to work with transcriptomic references only. Hence, RiboFlow references solely come from mRNA sequences. This is an important design choice with obvious limitations that the user needs to consider.

2.2 Ribo file

The output of RiboFlow is a binary file named ‘ribo’ which is built on top of the hierarchical data format (HDF) (The HDF Group, 1997-2019). Every ribo file must contain three types of data (Fig. 1B).

1. **Transcriptome annotation:** For each transcript in a given transcriptome, a ribo file contains their region annotations (5′ UTR, CDS, 3′ UTR), transcript names and lengths.
2. **Region RPF counts:** Ribosome profiling data are quantified using the number of reads mapping to the different transcript regions, namely the 5′ UTR, CDS and 3′ UTR. The distribution of reads across these regions can be informative as a quality control metric and depend on the RNase of choice in the experiment (Miettinen and Björklund, 2015; Wolin and Walter, 1988).
3. **Metagene counts:** Read counts are aggregated across all transcripts with respect to the translation start/stop sites. This data summarization is typically referred to as ‘metagene counts’.

The following data are optionally included in a ribo file:

RNA-Seq quantification: Most ribosome profiling experiments employ matched total RNA sequencing (RNA-Seq) to enable analyses of translation efficiency. RNA-seq quantifications are stored in a manner that parallels the region counts for the ribosome profiling experiment.

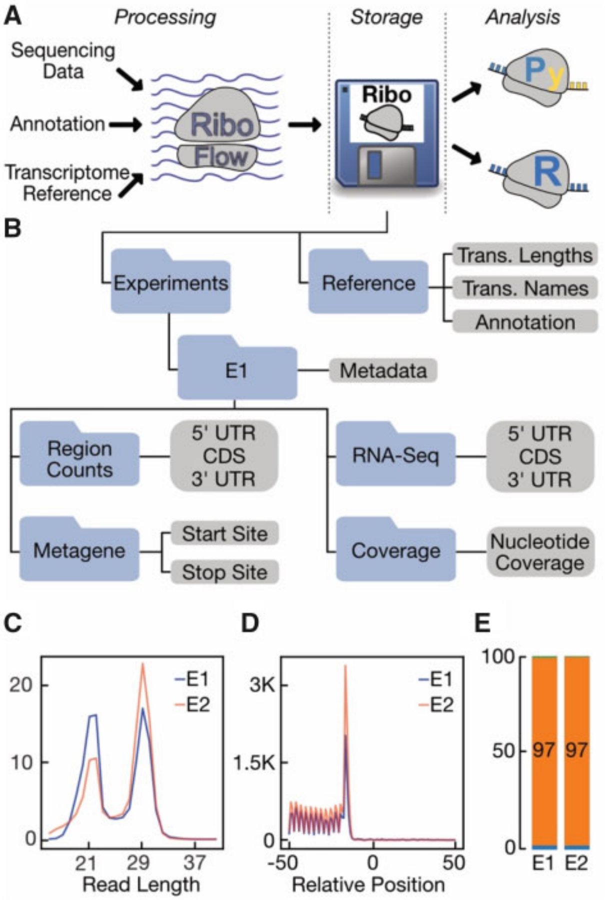

Fig. 1. (**A**) Overview of the Ribo Software Ecosystem. (**B**) The internal structure of a ribo file is depicted. (**C–E**) Example plots generated by RiboPy from two experiments denoted as E1 and E2. (C) Read length distribution of coding region mapping reads was plotted. The *y*-axis corresponds to the percentage of each read length. (D) RPFs mapping to the vicinity of the translation stop site are aggregated across genes to display a metagene plot. The *y*-axis corresponds to reads per million. (E) Percentage of reads mapping to different transcript regions are shown. Top (green) and bottom (blue) regions account for a small fraction of the data as they represent the 3′ and 5′ UTR mapping RPFs, respectively. Middle bar (orange) depicts the percentage of CDS mapping RPFs

Coverage data: Coverage data refers to the number of reads whose 5′ ends map to each nucleotide position for every transcript.

Metadata: A ribo file may contain metadata for any experiment or for the entire ribo file itself. Metadata is defined by the user on a key-value basis.

As a case study, we processed ∼7.3 billion reads across 58 ribosome profiling experiments from three studies (Cenik *et al.*, 2015; Sidrauski *et al.*, 2015; Wu *et al.*, 2019) (GEO accession numbers: GSE65912, GSE65778, GSE115162, respectively). For all three datasets, ribo files yielded more than 15-fold reduction in file size compared with gzip compressed text files (see Supplementary Material). For example, a ribo file containing all the above-described data from 50 experiments generated in Cenik *et al.* (2015) was only 110 MB in contrast to 1711 MB when using gzip compressed text files.

2.3 RiboR and RiboPy

To interact with ribo files, we offer an R package (RiboR) and a Python interface (RiboPy). RiboR and RiboPy provide a set of functions for data import into an R or Python environment and for commonly used visualization. In one function call, users can read ribosome occupancy around the start or stop sites (metagene data), or total read counts for a given transcript region (5′ UTR, CDS or 3′ UTR). Data can be aggregated or accessed for each individual transcript given a range of RPF lengths. Optional data such as metadata, transcript abundance and ribosome occupancy at nucleotide resolution can be obtained in a similar fashion. Finally, the built-in functions can generate visualizations for RPF length distribution, metagene data and region-specific read counts (Fig. 1C–E).

3 Conclusions

We describe the first specialized binary file format designed for ribosome profiling data. Storing data in ribo files not only reduces file sizes significantly but also facilitates efficient organization, portability and data analyses. We developed two interfaces, RiboR and RiboPy, for the most commonly used programming languages in bioinformatics, R and Python. RiboFlow allows users to process their raw sequencing data to generate ribo files across a wide spectrum of computational platforms, ranging from personal computers to high performance computing clusters. For further convenience and reproducibility, software installation and management are handled by package managers and container images. Given those features, this ecosystem offers a useful platform to study ribosome profiling data and provides superior user experience.

Acknowledgements

The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing high performance computing and storage resources that have contributed to the research results reported within this article. URL: http://www.tacc.utexas.edu. We also would like to thank Muxin Wang for her support in building filter sequence files, Xiaoying Wei for software testing and generation of annotation files for mouse, and William Zhang for software testing and processing raw data from Cenik *et al.* (2015).

Funding

This work was supported in part by the National Institutes of Health [CA204522 to C.C.] and Cancer Prevention and Research Institute of Texas [RR180042 to C.C.].

Conflict of Interest: none declared.

References

Abdennur,N. and Mirny,L. (2019) Cooler: scalable storage for Hi-C data and other genomically-labeled arrays. Bioinformatics, **36**, 311–316.

Arpat,A.B. *et al.* (2019) Transcriptome-wide sites of collided ribosomes reveal principles of translational pausing. bioRxiv 710061, doi: 10.1101/710061.

Berg,J.A. *et al.* (2019) XPRESSyourself: enhancing and automating the ribosome profiling and RNA-Seq analysis toolkit. bioRxiv, 704320, doi: 10.1101/704320

Birkeland,Å. *et al.* (2018) Shoelaces: an interactive tool for ribosome profiling processing and visualization. *BMC Genomics*, **19**, 543.

Carja,O. *et al.* (2017) riboviz: analysis and visualization of ribosome profiling datasets. *BMC Bioinformatics*, **18**, 461.

Cenik,C. *et al.* (2015) Integrative analysis of RNA, translation, and protein levels reveals distinct regulatory variation across humans. *Genome Res*, **25**, 1610–1621.

Chung,B.Y. *et al.* (2015) The use of duplex-specific nuclease in ribosome profiling and a user-friendly software package for Ribo-seq data analysis. *RNA*, **21**, 1731–1745.

Di Tommaso,P. *et al.* (2017) Nextflow enables reproducible computational workflows. *Nat. Biotechnol.*, **35**, 316–319.

Durand,N.C. *et al.* (2016) Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. *Cell Syst.*, **3**, 95–98.

Guydosh,N.R. and Green,R. (2014) Dom34 rescues ribosomes in 3’ untranslated regions. *Cell*, **156**, 950–962.

Guydosh,N.R. and Green,R. (2017) Translation of poly(A) tails leads to precise mRNA cleavage. *RNA*, **23**, 749–761.

Ingolia,N.T. *et al.* (2009) Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *Science*, **324**, 218–223.

Ingolia,N.T. *et al.* (2011) Ribosome profiling of mouse embryonic stem cells reveals the complexity and dynamics of mammalian proteomes. *Cell*, **147**, 789–802.

Ingolia,N.T. *et al.* (2019) Ribosome profiling: global views of translation. *Cold Spring Harb. Perspect. Biol.*, **11**: a032698.

Kristensen,A.R. *et al.* (2013) Protein synthesis rate is the predominant regulator of protein expression during differentiation. *Mol. Syst. Biol.*, **9**, 689.

Lareau,L.F. *et al.* (2014) Distinct stages of the translation elongation cycle revealed by sequencing ribosome-protected mRNA fragments. *Elife*, **3**, e01257.

Liakath-Ali,K. *et al.* (2018) An evolutionarily conserved ribosome-rescue pathway maintains epidermal homeostasis. *Nature*, **556**, 376–380.

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Ly,T. *et al.* (2014) A proteomic chronology of gene expression through the cell cycle in human myeloid leukemia cells. *Elife*, **3**, e01630.

Marguerat,S. *et al.* (2012) Quantitative analysis of fission yeast transcriptomes and proteomes in proliferating and quiescent cells. *Cell*, **151**, 671–683.

Melsted,P. *et al.* (2019) The barcode, UMI, set format and BUStools. *Bioinformatics*, **35**, 4472–4473.

Miettinen,T.P. and Björklund,M. (2015) Modified ribosome profiling reveals high abundance of ribosome protected mRNA fragments derived from 3’ untranslated regions. *Nucleic Acids Res.*, **43**, 1019–1034.

Perkins,P. *et al.* (2019) RiboStreamR: a web application for quality control, analysis, and visualization of Ribo-seq data. *BMC Genomics*, **20**, 422.

Popa,A. *et al.* (2016) RiboProfiling: a bioconductor package for standard Ribo-seq pipeline processing. *F1000Research*, **5**, 1309.

Schwanhäusser,B. *et al.* (2011) Global quantification of mammalian gene expression control. *Nature*, **473**, 337–342.

Sidrauski,C. *et al.* (2015) The small molecule ISRIB reverses the effects of eIF2α phosphorylation on translation and stress granule assembly. *Elife*, **4**,

The HDF Group (1997–2020) Hierarchical Data Format, version 5.

Wang,H. *et al.* (2019) Computational resources for ribosome profiling: from database to Web server and software. *Brief. Bioinform.*, **20**, 144–155.

Wolin,S.L. and Walter,P. (1988) Ribosome pausing and stacking during translation of a eukaryotic mRNA. *EMBO J.*, **7**, 3559–3569.

Wu,C.C.-C. *et al.* (2019) High-resolution ribosome profiling defines discrete ribosome elongation states and translational regulation during cellular stress. *Mol. Cell*, **73**, 959–970.e5.