The seeker R package: simplified fetching and processing of transcriptome data

Joshua L. Schoenbachler[1] and Jacob J. Hughey[1,2]

[1] Department of Biomedical Informatics, Vanderbilt University Medical Center, Nashville, Tennessee, United States
[2] Department of Biological Sciences, Vanderbilt University, Nashville, Tennessee, United States

ABSTRACT

Transcriptome data have become invaluable for interrogating biological systems. Preparing a transcriptome dataset for analysis, particularly an RNA-seq dataset, entails multiple steps and software programs, each with its own command-line interface (CLI). Although these CLIs are powerful, they often require shell scripting for automation and parallelization, which can have a high learning curve, especially when the details of the CLIs vary from one tool to another. However, many individuals working with transcriptome data are already familiar with R due to the plethora and popularity of R-based tools for analyzing biological data. Thus, we developed an R package called seeker for simplified fetching and processing of RNA-seq and microarray data. Seeker is a wrapper around various existing tools, and provides a standard interface, simple parallelization, and detailed logging. Seeker's primary output—sample metadata and gene expression values based on Entrez or Ensembl Gene IDs—can be directly plugged into a differential expression analysis. To maximize reproducibility, seeker is available as a standalone R package and in a Docker image that includes all dependencies, both of which are accessible at https://seeker.hugheylab.org.

Subjects Bioinformatics, Genomics, Data Science
Keywords Transcriptome data, Automation, Command-line interface, Genomics, RNA-seq, Microarray

Submitted 1 September 2022
Accepted 19 October 2022
Published 7 November 2022

Corresponding author
Jacob J. Hughey,
jakejhughey@gmail.com

Academic editor
Hong-Wei Sun

Additional Information and Declarations can be found on page 7

DOI **10.7717/peerj.14372**

INTRODUCTION

Measuring the transcriptome has become a standard approach for understanding biological systems. However, even after a transcriptome dataset is generated, it must go through multiple processing steps before it is ready for analysis. For RNA-seq data in particular, each step typically involves a different software program with its own command-line interface (CLI). Thus, complete processing of transcriptome data requires chaining together the output of one program with the input of another. Automating and parallelizing the entire process has often involved shell scripting, which can have a steep learning curve, limiting reproducibility and portability. To address this issue and simplify RNA-seq processing pipelines, two programs—pyrpipe (*Singh et al., 2021*) and nf-core/rnaseq (*Ewels et al., 2020*)—have recently been developed that wrap around various RNA-seq processing programs. pyrpipe uses Python, whereas nf-core/rnaseq uses Nextflow. However, many individuals working with transcriptome data are already familiar with R, given the popularity of R-based tools for biological data analysis, especially

for quantifying differential expression. Thus, we developed an R package called seeker, which simplifies the fetching and processing of RNA-seq and microarray data. Seeker provides an R-based interface for various existing programs and enables simple automation and parallelization.

METHODS

The seeker package's website (https://seeker.hugheylab.org) includes instructions for installing the package, multiple vignettes, and detailed documentation for every function.

System dependencies

For convenience, the seeker package includes a function called installSystemDeps that can install and configure the package's system dependencies: the NCBI SRA Toolkit (which includes the tools prefetch and fasterq-dump; https://github.com/ncbi/sra-tools), Miniconda (with Python 3; https://docs.conda.io/en/latest/miniconda.html), the Mamba package manager (https://github.com/mamba-org/mamba), and the conda packages fastq-screen (*Wingett*), fastqc (*Andrews*), multiqc (*Ewels et al., 2016*), pigz, refgenie (*Stolarczyk et al., 2020*), salmon (*Patro et al., 2017*), and trim-galore (*Krueger*). installSysDeps sets environmental variables so the CLIs are accessible within R, and installs snakemake (*Mölder et al., 2021*) to make it easier to use seeker in reproducible analyses. We also provide a Docker image called socker, based on rocker/rstudio from the Rocker Project, in which seeker and its dependencies are already installed.

The seeker package has no hardware requirements other than those of its system dependencies and of R itself. In general, processing RNA-seq data is more memory- and compute-intensive than processing microarray data. However, the memory requirements and computation time to process a given RNA-seq dataset will depend on one's hardware, the number of files to process and their size, the extent of parallelization, and which processing steps are being run. For specific estimates, we refer readers to the papers and documentation of the underlying tools.

Microarray data

To fetch and process microarray data, the package includes a function called seekerArray (Fig. 1A), which depends on the GEOquery and ArrayExpress R packages. seekerArray can process data from NCBI GEO or Array Express, or raw Affymetrix data stored locally. The main inputs to seekerArray are a study accession and a preferred gene ID type (Entrez or Ensembl). The main outputs are a table of sample metadata and a matrix of $\log_2$-transformed gene expression measurements. seekerArray automatically detects the microarray platform and maps probes to the appropriate gene IDs. If the study includes raw Affymetrix data, seekerArray performs RMA (*Irizarry, 2003*) and uses custom CDFs from Brainarray (*Dai, 2005*), otherwise seekerArray uses the study's processed data.

RNA-seq data

To fetch and process RNA-seq data, the package includes multiple functions (described below), each of which handles single-end or paired end reads, includes sensible defaults, and allows the user to pass custom arguments to the respective APIs or CLIs.

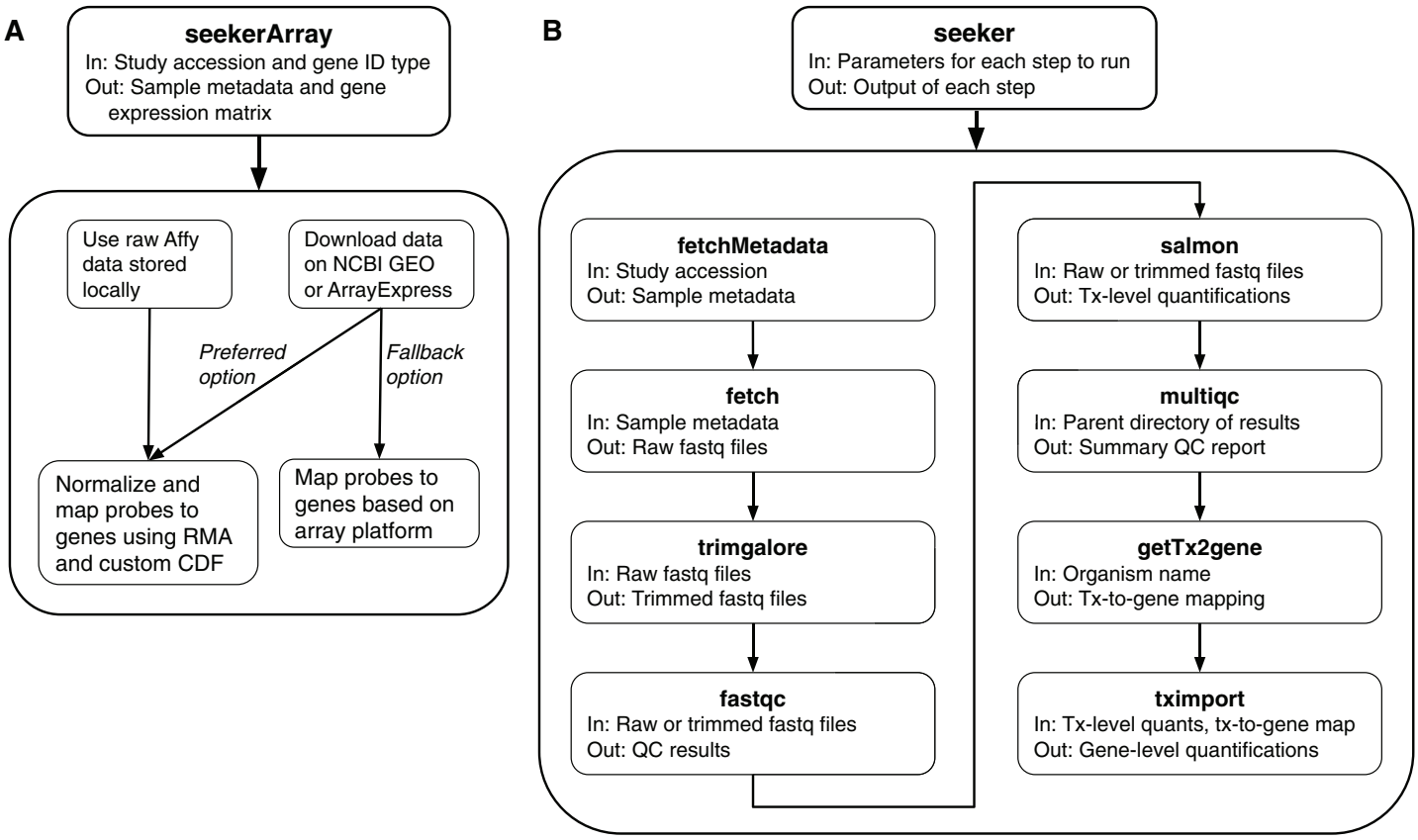

Figure 1 Schematics of the seekerArray and seeker functions of the seeker R package. (A) The preferred option requires raw Affymetrix data. The fallback option uses processed data. (B) The main inputs and outputs are listed for each function. Full-size DOI: 10.7717/peerj.14372/fig-1

- fetchMetadata: Uses the API of the European Nucleotide Archive (ENA) or the Sequence Read Archive (SRA) to fetch a study's sample metadata. The main input is a Bioproject accession.
- fetch: Uses srafetch, fasterq-dump, and pigz to download files from SRA and convert them to gzipped fastq files in parallel. The main input is a vector of SRA run accessions.
- trimgalore: Uses Trim Galore to perform standard adapter and quality trimming in parallel. The main input is a vector of paths to fastq files.
- fastqc: Uses FastQC to perform quality control checks in parallel. The main input is a vector of paths to fastq files.
- salmon: Uses Salmon to quantify transcript abundances in parallel. The main inputs are a vector of paths to fastq files, corresponding sample names, and a directory containing the salmon transcriptome index (which can be fetched by refgenie *via* installSysDeps).
- multiqc: Uses MultiQC to aggregate the results of various processing steps (including Trim Galore, FastQC, and Salmon) into a single report. The main input is a directory containing the results.
- getTx2gene: Uses the biomaRt package (*Durinck et al., 2009*) to create a mapping of transcripts to genes based on Ensembl IDs. The main input is an organism name.

- tximport: Uses the tximport R package (*Soneson, Love & Robinson, 2015*) to summarize Salmon's transcript-level quantifications for gene-level analyses. The main inputs are a directory containing quantification directories from Salmon and a mapping of transcripts to genes.

The most computationally intensive functions (fetch, trimgalore, fastqc, and salmon) output their progress as tab-delimited text files.

In addition, the package includes a function called seeker, which can perform any of the above steps and pass the output of one step as input to the next step (Fig. 1B). The main input to the seeker function is a list of parameters, which can be derived from a yaml file, that specifies which steps to perform and how to perform them. Depending on the steps specified, the seeker function can fetch and process publicly available RNA-seq data or process locally stored data. The function also includes a "dry run" option to check the parameters' validity without fetching or processing any data.

Reproducibility

To help ensure that the output of seekerArray and seeker is reproducible, both functions save a yaml file of the user-defined parameters and a text file containing the R session information. The latter is provided by the sessioninfo R package and includes version numbers and sources of all loaded packages. In addition, the seeker function saves a text file containing paths and version numbers of its system dependencies, as well as a yaml file of the conda environment, which includes version numbers of the miniconda-based dependencies.

To provide an even higher level of reproducibility, the seeker package and its dependencies are available in a Docker image called socker (https://github.com/hugheylab/socker), which is based on rocker/rstudio (https://hub.docker.com/r/rocker/rstudio). Using socker thus ensures that the output is not dependent on details of the local computing environment (*Nüst et al., 2020*).

RESULTS

To illustrate the utility of the seeker package, we used it to fetch and process multiple publicly available transcriptome datasets. We first used the seekerArray function to fetch two datasets related to circadian metabolism in mouse liver (GSE34018 and GSE67964) (*Cho et al., 2012*; *Zhang et al., 2015*). The main output for GSE34018, which is based on an Illumina beadchip, consisted of 24 samples and 17,142 Entrez genes. The main output for GSE67964, which is based on an Affymetrix array, consisted of eight samples and 27,352 Entrez genes. We used these outputs to perform principal components analysis and to calculate differential expression using limma (*Ritchie et al., 2015*) between the wild-type and knockout samples in each dataset (Fig. 2). The $\log_2$ fold-changes for Entrez genes measured in the two datasets were weakly negatively correlated (Spearman's rho −0.092), consistent with the opposing roles of the genes knocked out in each dataset (Nr1d1 and Nr1d2 in GSE34018; Rora and Rorc in GSE67964).

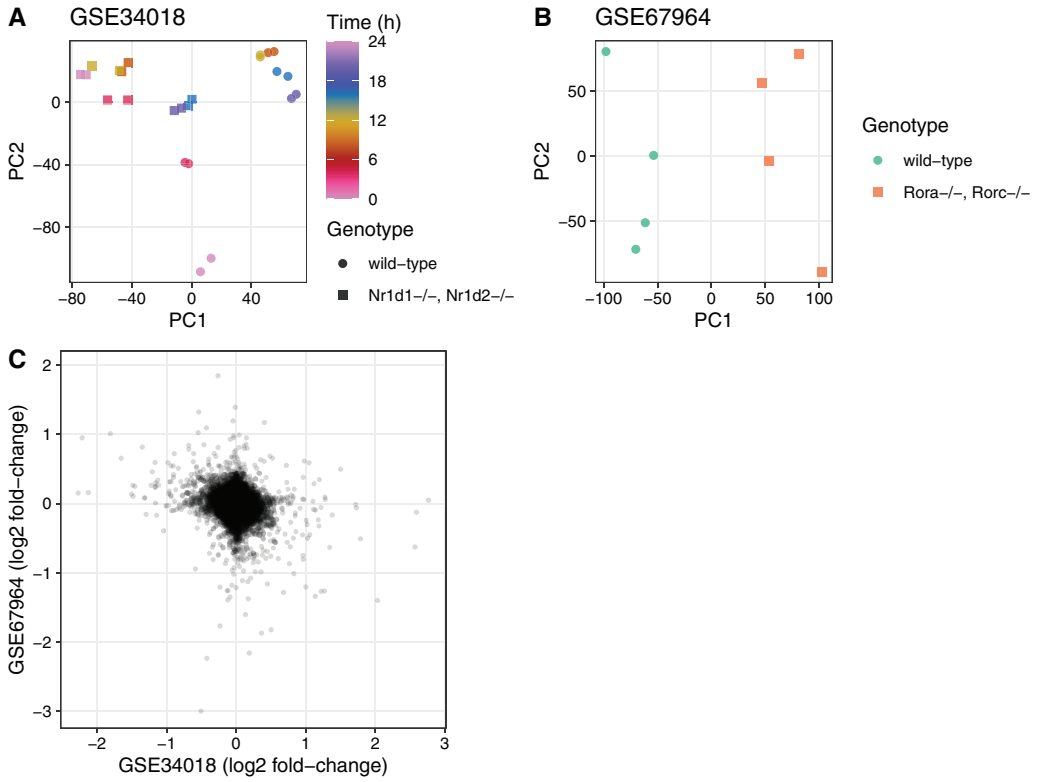

Figure 2 Example of using the output of seekerArray. Principal components analysis of (A) GSE34018 and (B) GSE67964, both of which are microarray datasets based on livers of mice in a 12 h:12 h light:dark cycle. Each point represents a sample. In (A), color indicates time since lights on and shape indicates genotype (the knockout was liver-specific). In (B), color and shape indicate genotype (all samples were collected at 22 h after lights on). (C) Scatterplot of $\log_2$ fold-changes of differential expression between wild-type and knockout for the 16,747 Entrez genes measured in both datasets. Each point represents a gene. Full-size DOI: 10.7717/peerj.14372/fig-2

We next used the seeker function to fetch and process one sample from each of two RNA-seq datasets related to circadian rhythms and feeding in mouse liver (PRJNA600892 and PRJNA667743) (*Guan et al., 2020*; *Manella et al., 2021*). We chose two samples that had the same genotype (wild-type), feeding regimen (*ad libitum*), and time of day of acquisition (4 h after lights on). PRJNA600892 is based on paired-end sequencing and Illumina TruSeq Stranded library prep, whereas PRJNA667743 is based on single-end sequencing and bulk MARS-Seq (3′-tagged) library prep (*Jaitin et al., 2014*). The seeker function detected and handled the paired-end and single-end reads automatically.
To account for the 3′-tagged reads in PRJNA667743, we specified the countsFromAbundance argument to tximport as "no" in the input parameters to seeker. For simplicity and speed, we disabled the trimgalore step and used a salmon index based on partial selective alignment. The main outputs were the sample metadata and the tximport object including counts and abundances for 35,494 Ensembl genes. As expected, the gene-level abundances between the samples from the two datasets were highly correlated (Fig. 3; Spearman's rho 0.877).

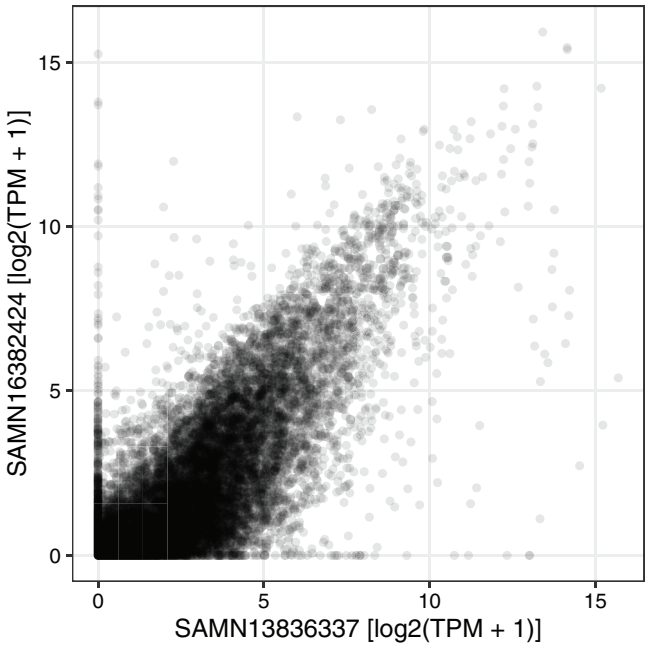

Figure 3 Example of using the output of seeker. Scatterplot of abundance for 35,494 Ensembl genes in sample SAMN13836337 (from PRJNA600892) and sample SAMN16382424 (from PRJNA667743). Each point represents a gene. TPM stands for transcripts per million.

Full-size DOI: 10.7717/peerj.14372/fig-3

DISCUSSION

The seeker package complements existing software packages for fetching and processing transcriptome data. Whereas pyrpipe and nf-core/rnaseq are focused on RNA-seq data, seeker can also process microarray data. Here seeker builds on the GEOquery and ArrayExpress R packages and our previous package metapredict (*Hughey & Butte, 2015*) by providing a simple interface to map microarray probe sets to standard gene IDs. In addition, unlike pyrpipe, seeker can run in parallel natively, without requiring a workflow manager such as Nextflow. seeker also allows the user to specify all processing parameters for a given dataset in a single yaml file. Perhaps most importantly, seeker uses the R language and computing environment, and thus could appeal to a wider group of researchers already using R for bioinformatic analyses.

The seeker package does have limitations. First, the current implementation focuses on processing bulk transcriptome data using a relatively small set of tools. Although these should suffice for the majority of use cases, seeker currently supports a more limited set than either pyrpipe or nf-core/rnaseq. For example, it supports only salmon for quantifying transcript-level abundances, which is a standard in the field and is both fast and accurate. In the future, we plan to extend seeker to accommodate other types of genomic data, including from single cells, and a wider range of tools. Second, although seeker thoroughly documents its computing environment, recreating that environment on another machine (if not using the socker Docker image) would still be a manual process. We welcome contributions from the community to improve seeker in these and any other

ways. By providing a straightforward, R-based approach to small- or large-scale processing of transcriptome data, we hope seeker contributes to improved reproducibility and transparency of genomic analyses.

ACKNOWLEDGEMENTS

Reproducible results are available at https://doi.org/10.6084/m9.figshare.20720848.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the National Institute of General Medical Sciences (R35GM124685). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
National Institute of General Medical Sciences: R35GM124685.

Competing Interests

Jacob J. Hughey is an Academic Editor for PeerJ.

Author Contributions

- Joshua L. Schoenbachler performed the experiments, analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Jacob J. Hughey conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The reproducible results are available at Figshare: Hughey, Jacob (2022): Reproducible results for: The seeker R package: simplified fetching and processing of transcriptome data. figshare. Dataset. https://doi.org/10.6084/m9.figshare.20720848.v2.

REFERENCES

Andrews S. 2020. FastQC: a quality control analysis tool for high throughput sequencing data. *Available at https://www.bioinformatics.babraham.ac.uk/projects/fastqc/*.

Cho H, Zhao X, Hatori M, Yu RT, Barish GD, Lam MT, Chong L-W, DiTacchio L, Atkins AR, Glass CK, Liddle C, Auwerx J, Downes M, Panda S, Evans RM. 2012. Regulation of circadian behaviour and metabolism by REV-ERB-α and REV-ERB-β. *Nature* **485(7396)**:123–127 DOI 10.1038/nature11048.

Dai M. 2005. Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data. *Nucleic Acids Research* **33(20)**:e175 DOI 10.1093/nar/gni179.

Durinck S, Spellman PT, Birney E, Huber W. 2009. Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. *Nature Protocols* **4(8)**:1184–1191 DOI 10.1038/nprot.2009.97.

Ewels P, Magnusson M, Lundin S, Käller M. 2016. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **32(19)**:3047–3048 DOI 10.1093/bioinformatics/btw354.

Ewels PA, Peltzer A, Fillinger S, Patel H, Alneberg J, Wilm A, Garcia MU, Di Tommaso P, Nahnsen S. 2020. The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology* **38(3)**:276–278 DOI 10.1038/s41587-020-0439-x.

Guan D, Xiong Y, Trinh TM, Xiao Y, Hu W, Jiang C, Dierickx P, Jang C, Rabinowitz JD, Lazar MA. 2020. The hepatocyte clock and feeding control chronophysiology of multiple liver cell types. *Science* **369(6509)**:1388–1394 DOI 10.1126/science.aba8984.

Hughey JJ, Butte AJ. 2015. Robust meta-analysis of gene expression using the elastic net. *Nucleic Acids Research* **43(12)**:e79 DOI 10.1093/nar/gkv229.

Irizarry RA. 2003. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* **4(2)**:249–264 DOI 10.1093/biostatistics/4.2.249.

Jaitin DA, Kenigsberg E, Keren-Shaul H, Elefant N, Paul F, Zaretsky I, Mildner A, Cohen N, Jung S, Tanay A, Amit I. 2014. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science* **343(6172)**:776–779 DOI 10.1126/science.1247651.

Krueger F. 2021. TrimGalore: a wrapper around Cutadapt and FastQC to consistently apply adapter and quality trimming to FastQ files, with extra functionality for RRBS data. *Available at https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/*.

Manella G, Sabath E, Aviram R, Dandavate V, Ezagouri S, Golik M, Adamovich Y, Asher G. 2021. The liver-clock coordinates rhythmicity of peripheral tissues in response to feeding. *Nature Metabolism* **3(6)**:829–842 DOI 10.1038/s42255-021-00395-7.

Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, Sochat V, Forster J, Lee S, Twardziok SO, Kanitz A, Wilm A, Holtgrewe M, Rahmann S, Nahnsen S, Köster J. 2021. Sustainable data analysis with Snakemake. *F1000Research* **10**:33 DOI 10.12688/f1000research.29032.2.

Nüst D, Sochat V, Marwick B, Eglen SJ, Head T, Hirst T, Evans BD, Markel S. 2020. Ten simple rules for writing Dockerfiles for reproducible data science. *PLOS Computational Biology* **16(11)**:e1008316 DOI 10.1371/journal.pcbi.1008316.

Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. 2017. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods* **14(4)**:417–419 DOI 10.1038/nmeth.4197.

Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, Smyth GK. 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research* **43(7)**:e47 DOI 10.1093/nar/gkv007.

Singh U, Li J, Seetharam A, Wurtele ES. 2021. pyrpipe: a Python package for RNA-Seq workflows. *NAR Genomics and Bioinformatics* **3(2)**:lqab049 DOI 10.1093/nargab/lqab049.

Soneson C, Love MI, Robinson MD. 2015. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research* **4**:1521 DOI 10.12688/f1000research.7563.2.

Stolarczyk Mł, Reuter VP, Smith JP, Magee NE, Sheffield NC. 2020. Refgenie: a reference genome resource manager. *GigaScience* **9(2)**:1760 DOI 10.1093/gigascience/giz149.

Wingett S. 2022. FastQ-Screen: detecting contamination in NGS data and multi-species analysis. *Available at https://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/*.

Zhang Y, Fang B, Emmett MJ, Damle M, Sun Z, Feng D, Armour SM, Remsberg JR, Jager J, Soccio RE, Steger DJ, Lazar MA. 2015. GENE REGULATION. discrete functions of nuclear receptor Rev-erbα couple metabolism to the clock. *Science* **348(6242)**:1488–1492 DOI 10.1126/science.aab3021.