Complexity of Transcriptomic Data Analysis and Implications for Biological Discovery

Thesis by

Laura Luebbert

In Partial Fulfillment of the Requirements for
the Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2024
(Defended March 14th, 2024)

Laura Luebbert
ORCID: 0000-0003-1379-2927

ACKNOWLEDGEMENTS

I dedicate this thesis to my family. To my mom, you fought for your education so I would not have to. To my dad, who always believed in me. To my sister Saskia, who will always be my baby sister even though she has far outgrown me and become my inspiration (I am so proud of you!). To my sister Annette, who has tirelessly supported my painting hobby and shows my niece and nephew pictures of me so they will recognize their aunt from across the Atlantic. To my brother Matthias, because I can dive to 100 feet with you and never feel unsafe. To my niece and nephews, Julian, Ilai, Jona, and Laila, please forgive me for not being around more. Per Albert, Paqui, Andreu i Patricia perquè Catalunya no seria casa meva sense vosaltres. Per els meus avis, Maria i Gerard, perquè sempre heu cregut en mi.

I would like to thank my advisor, Lior Pachter, for his relentless support and his trust and confidence in me. Thanks to you, I rediscovered my love for science and dared to dive into a new field. Our accomplishments reflect how wonderful it was to work with you. I would also like to thank you for being incredible at selecting your team: Ángel, Anne, Ca(ke)t, Delaney, Gennady, Joseph, Kayla, Kristján, Lambda, Maria, Mayuko, Meichen, Nikki, Rebekah, Taleen, and Tara. Thank you all for becoming my Caltech family!

I thank my committee, David van Valen, Pamela Bjorkman, and Richard Murray, for their advice and support. I would also like to thank Henry Lester, who first introduced me to Caltech, Aaron Nichols, who, with inexhaustible patience, guided me through my first serious wet-lab experiments, and Saul Priceman, who trusted me to analyze his incredible dataset and has become a valued mentor. I would like to thank my first mentor, Dennis Claessen, who has supported me my entire professional career – the last 10 (!!!) years.

I would like to thank Victoria Jorgensen, who immediately and without hesitation took me under her wing and showed me the ways of the USA after I arrived here alone and scared. I would not have made it through my Ph.D. without you.

I was extremely fortunate to have been surrounded by the most supportive, fun, loving, and brilliant people throughout my Ph.D., all of whom I do not have the space to mention here. However, I want to give a shout-out to Zsófia Török ("The flower that blooms in adversity is the most rare and beautiful of all." - Mulan), Pranav Bhamidipati (for long pandemic phone calls that evolved into long post-pandemic walks), Matteo Guareschi (qualcosa in italiano), Victor Garcia-Ruiz (te aprecio cabron), Aditi Narayanan (the best Laugenbrötchen), Tom Röschinger (kann ja nicht jeder in Leverkusen aufwachsen), Tom Naragon (because you can never have too many Toms in your life), Tyler Ross (not the bees!!), Valentijn Broeken (die helemaal vanuit Switzerland op bezoek kwam), and Ha-Lan Tran (die mij Nederlands en laboratorium technieken tegelijkertijd leerde).

Finally, I would like to thank Thomas Henning for all the love and laughter, for being my #1 fan, for grumpily following me up mountains at 6 am, and for a lifetime supply of Trader Joe's Jellybean Hearts, which I will probably finish before this thesis is done.

I love you all.

ABSTRACT

Over the past decade, the advancement of ‘omics’ technologies has ushered in a new era for the life sciences. Given the high-throughput nature of omics technologies, this era is characterized by unique computational challenges pertaining to data size and dimensionality, and technical and biological noise. Concurrently, it offers opportunities, as global, untargeted, and parallel measurement of large amounts of information often captures unexpected insights.

This thesis describes challenges inherent to the omics era of life sciences, particularly highlighting the increasing importance of merging expertise in biology and computer science. It describes the development of multiple software tools designed to address several of these challenges, which were immediately adopted and widely implemented in transcriptomics and proteomics research. Additionally, it contains three chapters focused on unraveling previously unquantifiable information, including the interpretation of sequencing data from organisms with low-quality reference genome assemblies and workflows for identifying novel viruses using single-cell RNA sequencing data already massively generated in research, healthcare, and agriculture.

PUBLISHED CONTENT AND CONTRIBUTIONS

Laura Luebbert, Delaney K. Sullivan, Maria Carilli, Kristján Eldjárn Hjörleifsson, Alexander Viloria Winnett, Tara Chari, Lior Pachter (2023). Efficient and accurate detection of viral sequences at single-cell resolution reveals novel viruses perturbing host gene expression. *bioRxiv*. https://doi.org/10.1101/2023.12.11.571168

> **LL** and LP conceived the project following the publication of the PalmDB database. **LL** and LP conceived of translated search (the --aa option) and LL implemented it in kallisto and kb-python with support from DKS and KEH. **LL**, DKS, and LP conceived of the host masking options, and LL implemented them with support from DKS. MC wrote the logistic regression models with help from **LL**. AVW provided data used for the validation of kallisto translated search. TC provided crucial guidance on statistical analyses. **LL** led and performed (unless otherwise attributed) the analyses described in the paper, created the figures, and wrote the initial draft of the manuscript. All authors reviewed and approved the manuscript.

Laura Luebbert*, Chi Hoang*, Manjeet Kumar, Lior Pachter (2023). Fast and scalable querying of eukaryotic linear motifs with *gget elm*. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btae095

> **LL** and LP conceived the project after listening to a lecture by Prof. Amy E. Keating. **LL**, CH, and MK designed the *gget elm* approach. **LL** and CH wrote the *gget elm* software, with CH being the primary developer under the supervision of **LL**. **LL** is the primary developer of the *gget* software, and MK is the primary developer of the ELM resource. **LL** wrote the initial draft of the manuscript. CH, MK, and LP provided feedback on the manuscript. All authors reviewed and approved the manuscript.

Laura Luebbert, Lior Pachter (2023). Efficient querying of genomic reference databases with *gget*. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btac836

> **LL** and LP conceived the project after **LL** recognized problems frequently encountered by biologists performing transcriptomic data analysis. **LL** wrote the *gget* software, the software documentation, and the initial draft of the manuscript. All authors reviewed and approved the manuscript.

Delaney K Sullivan, Kyung Hoi Min, Kristján Eldjárn Hjörleifsson, **Laura Luebbert**, Guillaume Holley, Lambda Moses, Johan Gustafsson, Nicolas L Bray, Harold Pimentel, A Sina Booeshaghi, Páll Melsted, Lior Pachter (2023). kallisto, bustools, and kb-python for quantifying bulk, single-cell, and single-nucleus RNA-seq. *bioRxiv*. https://doi.org/10.1101/2023.11.21.568164

All authors contributed either directly to kallisto, bustools, kb-python, or to the methods implemented in the software. DKS led the development of the latest versions (at the time of writing this manuscript) of kallisto (version 0.50.1), bustools (version 0.43.1), and kb-python (version 0.28.0). NLB conceived kallisto. ASB conceived kb-python and KMH created, implemented, and developed kb-python under the supervision of ASB. LM identified the need for creating a transcriptome FASTA file coherent with a genome and GTF as implemented in kb-python. ASB and PM implemented the initial version of bustools and its interface with kallisto, which was published in Melsted, Booeshaghi et al., 20211 where early versions of these software were benchmarked. DKS and KEH implemented the d-list option in kallisto and adapted kallisto to use the Bifrost de Bruijn graph with help from GH. **LL** conceived the translated search (the --aa option) and implemented it with help from DKS and KEH, and extensively tested the latest versions of kallisto, bustools, and kb-python. JG augmented the functionalities of bustools. DKS refactored kallisto providing additional modularity with respect to the expectation-maximization algorithm implemented by HP, and unifying the treatment of bulk and single-cell data. PM and LP supervised the initial development and coordination of kallisto and bustools. DKS drafted the initial manuscript. All authors edited and reviewed the final manuscript.

Zsofia Torok, **Laura Luebbert**, Jordan Feldman, Alison Duffy, Alexander A. Nevue, Shelyn Wongso, Claudio V. Mello, Adrienne Fairhall, Lior Pachter, Walter G. Gonzalez, Carlos Lois (2023). Recovery of a learned behavior despite partial restoration of neuronal dynamics after chronic inactivation of inhibitory neurons. *bioRxiv*. https://doi.org/10.1101/2023.05.17.541057

ZT and CL conceived the experimental design of the study. ZT performed the behavioral, single-cell, and electrophysiology experiments and data collection. ZT, **LL**, SW, AAN, JP designed and performed the data analysis and interpretation. **LL** designed and performed the analysis and interpretation of the transcriptomic data. ZT, **LL**, JF, AD, and WG wrote the initial draft of the manuscript. All authors edited and reviewed the final manuscript.

Tanya B. Dorff, M. Suzette Blanchard, Lauren N. Adkins, **Laura Luebbert,** Neena Leggett, Stephanie N. Shishido, Alan Macias, Marissa Del Real, Gaurav Dhapola, John P. Murad, Ammar Chaudhry, Hripsime Martirosyan, Ethan Gerdts, Jamie R. Wagner, Tracey Stiller, Dileshni Tilakawardane, Sumanta Pal, Catalina Martinez, Elizabeth L. Budde, Massimo D'Apuzzo, Peter Kuhn, Lior Pachter, Stephen J. Forman, Saul J. Priceman (2024). PSCA-CAR T cells for metastatic castration-resistant prostate cancer: First-in-human phase 1 trial. *Under review.*

TBD, MSB, LNA, **LL**, NL, SNS, AM, MDR, GD, JPM, AC, HM, EG, JRW, TS, DT, SP, RER, CM, ELB, MD, PK, LP, SJF, and SJP participated in the design, execution, and data analysis and/or interpretation of the reported results. TBD, MSB, LNA, **LL**, NL, SNS, AM, MDR, GD, JPM, MD, PK, LP, SJF, and SJP contributed to acquisition of, and analysis of, data. **LL**, NL, and LP designed, executed and interpreted the single-cell RNA sequencing data analysis. SJP and TBD wrote the manuscript. TBD, SJF, and SJP supervised all aspects of the study. All authors reviewed, edited, and/or advised on the manuscript.

* Contributed equally

TABLE OF CONTENTS

Chapter 1

INTRODUCTION – PART I

From Beaker to (Peta)Byte

Over the past decade, the 'omics' era of the life sciences has led to a significant increase in the volume and complexity of biological data. The Sequence Read Archive, which stores raw sequencing data and alignment information, has grown to >30 petabases since its establishment in 2012[1]. In 1996, the NCBI GenBank, which stores annotated DNA sequences, was sent to subscriber's homes in the format of 7 CDs (Figure 1.1). Today, GenBank contains 2.5 billion sequences, and one would require 5,703 CDs (assuming a capacity of 700 MB per CD) to follow GenBank's original distribution model (the GenBank Release 258.0 (https://www.ncbi.nlm.nih.gov/genbank/release/258) requires roughly 3,992 GB of disk space). To store the Sequence Read Archive on CDs, one would require upwards of 42,857,143 CDs. To tackle the increasing size, as well as the heterogenicity and noisiness of omics data, a myriad of software programs continues to be released daily.

As a wet-lab geneticist who learned how to code and switched to the field of computational biology during her Ph.D., I noticed that there were often overlooked yet crucial factors beyond how well a software program performs its designated function that contribute heavily to whether a program will be widely implemented. I incorporated these into the software programs described in this thesis, which were instantly adopted and became a worldwide standard in the analysis of transcriptomic and proteomic data analysis (Figure 1.2). Here, I will describe and quantify some of these factors.

Figure 1.1 Photograph of the CDs containing release 97.0 of the NCBI GenBank database as distributed to subscribers in 1996. Courtesy of Prof. Lior Pachter.

I will use the software tools documented in the scRNA-tools database[2] (https://www.scrna-tools.org/) to model the current state of software released to support omics research. Single-cell RNA sequencing (scRNA-seq) is a method to measure all RNA molecules in thousands of individual cells in parallel while retaining single-cell resolution, and it is one of the most widely used omics technologies that emerged over the past decade. According to the scRNA-tools database, 1,706 software programs were released between September 2016 and January 2024 for the analysis of scRNA-seq data – approximately one tool every second day. 107 of these tools were not published. To get an estimate of the extent to which the tools were used in practice, Figure 1.3 shows the number of citations of tools that

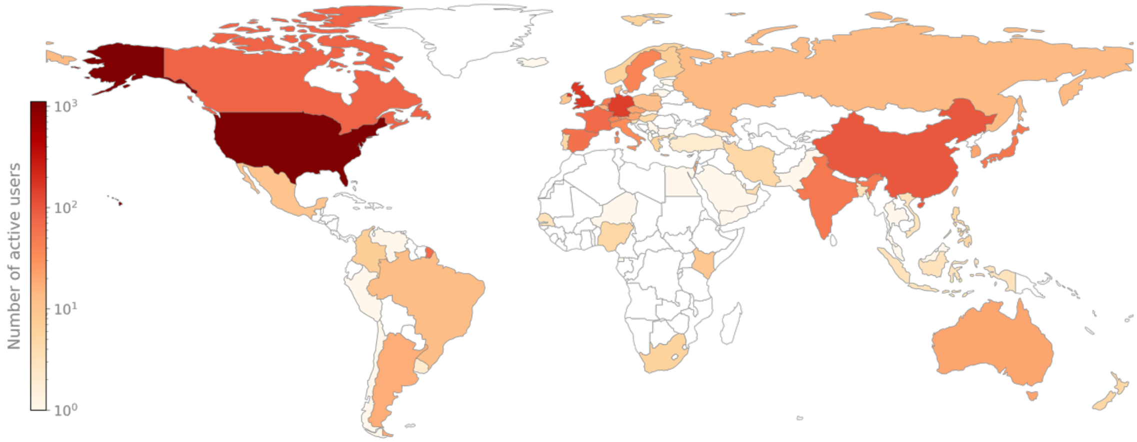


Figure 1.2 The number of active users of the *gget* website (https://pachterlab.github.io/gget/) by country between November 2023 and February 2024. The code to reproduce this figure can be found here: https://github.com/lauraluebbert/lauraluebbert.

were published between September 2016 and December 2022 (to allow at least one year to gather citations). I note that this method will be biased towards tools that have had more time to accumulate citations. Only 35 (3 %) of published tools received over 1,000 citations, suggesting extensive use, with the most highly cited tools STAR[3], Seurat[4], Monocle[5], Salmon[6], and kallisto[7] having been cited 32,005, 29,316, 7,912, 7,030, and 7,011 times, respectively. 490 (46 %) of published tools received less than 20 citations. This indicates that almost half of peer-reviewed software programs published for analyzing scRNA-seq data are barely used in practice. To understand the difference between tools that end up being widely used and those that don't, we first need to understand the user base.

Omics data is highly complex, both in terms of the computational requirements of its analysis and the underlying biological implications. To rigorously analyze and interpret high-dimensional omics data, extensive knowledge in both computer science and biology is required. However, only recently have undergraduate and graduate biology programs begun to include advanced programming classes in their curriculums, and many biology students still begin their Ph.D. and PostDoc positions with no to minimal coding skills[8,9]. Hence, widely used omics tools need to accommodate novice programmers. From my experience as I evolved from novice programmer to writing software for novice and advanced bioinformaticians, there are three major obstacles an omics software's user interface needs to overcome for successful, widespread, and long-lived implementation: installation, documentation, and updates.

Installation

The first hurdle when using a new software program is the installation. There are several program repositories and associated package managers that greatly simplify the installation of software programs. For Python programs, the most widely used program repositories for omics software are *PyPI* and *Bioconda*. Both allow the installation of software in a single line of code, greatly simplifying the process compared to requiring users to run a container

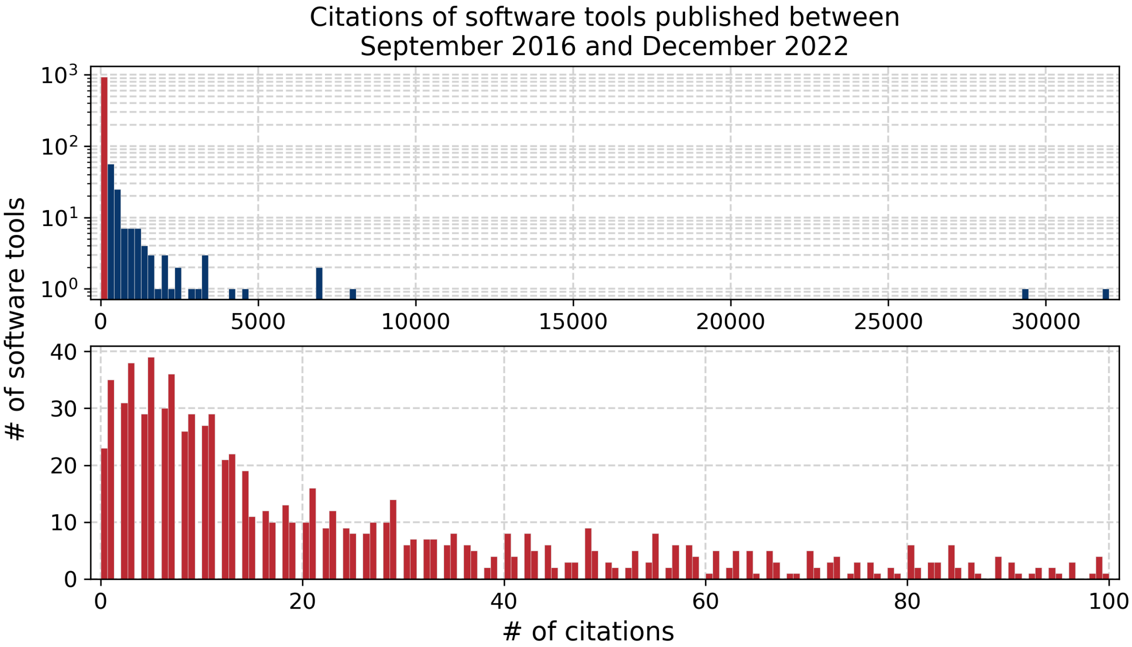


Figure 1.3 The number of citations for software tools used in the analysis of scRNA-seq data published between September 2016 and December 2022 according to the scRNA-tools database (https://www.scrna-tools.org/). The top plot shows the histogram for all published tools. The first bin in the top plot consists of tools with 0-100 citations and is broken down further in the bottom plot. The code to reproduce this figure can be found here: https://github.com/lauraluebbert/PhD_thesis/blob/main/Chapter1_Introduction.ipynb.

application, e.g., through Docker, or compile the software from source code. Even without testing whether the installation is functional, highly cited Python tools are more likely to have available PyPI and/or Bioconda installations (Figure 1.4A).

Documentation

Next, the user needs to learn how to use the newly installed software program. Ideally, software documentation is provided in the form of Python/R function descriptions, shell script help arguments, a GitHub README and/or wiki, and/or a separate documentation website. All 35 published tools with over 1,000 citations in the scRNA-tools database provide an extensive, publicly available manual containing software documentation, installation guidelines, quick start guides, and tutorials (Table 1.1). Except for the programs BackSPIN, CellChat, DoubletFinder, Scrublet, and MAGIC, for which documentation and tutorials are included in the GitHub README, these manuals are hosted on a website separate from the GitHub code repository. The Bioconductor project has set a commendable example by requiring contributors to adhere to a minimum standard of guidelines outlining, amongst others, package documentation (contributions.bioconductor.org). The accessibility of the documentation can be further increased by following Americans with Disabilities Act (ADA) guidelines for web content, as well as providing translations to different languages. Based on Google Analytics tracking of the documentation website for the software tool *gget*[10] (https://pachterlab.github.io/gget/), further described in Chapter 2, the number of Spanish-speaking users increased by 35 %

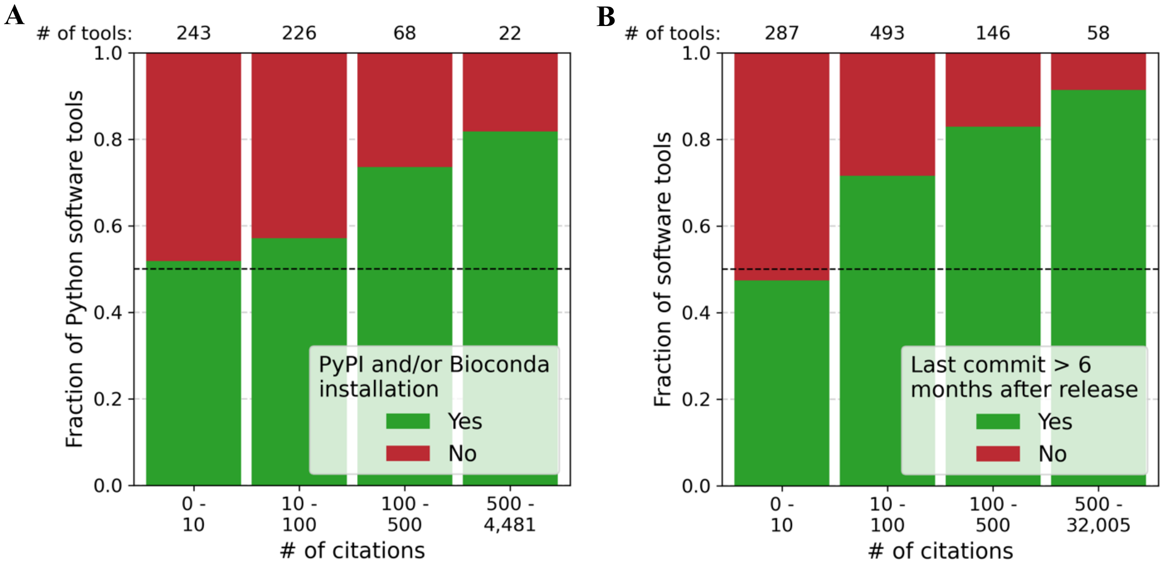


Figure 1.4 **A** Fraction of published Python software tools in the scRNA-tools database (https://www.scrna-tools.org/) with available installations from PyPI or Bioconda binned by number of citations. **B** Fraction of software tools in the scRNA-tools database (https://www.scrna-tools.org/) released between September 2016 and December 2022 for which the last GitHub commit was at least 6 months after the initial software release binned by number of citations. The code to reproduce these figures can be found here: https://github.com/lauraluebbert/PhD_thesis/blob/main/Chapter1_Introduction.ipynb.

(from an average 3.1 to 4.2 new users per month) after the *gget* documentation was also made available in Spanish.

Updates

With the pressure to produce and publish in academic research, it may be tempting to move on to the next project after the release of a software program and forget all about the latter. However, as omics methods evolve, data structures and package dependencies are likely to be updated or changed, and programs must also evolve with these updates to provide continued usability. Figure 1.4B shows the fraction of tools in the scRNA-tools database released between September 2016 and December 2022, for which the latest GitHub commit was at least six months after the initial release of the software tool. Using the latest GitHub commit as an indicator for a software update, 91.4 % of highly cited (>500 citations) software programs were updated after the initial release compared to 47.4 % of less cited (0-10 citations) programs (Figure 1.4B).

The following subchapter lists specific guidelines for user-friendly omics technologies based on the factors discussed here. User-friendliness is especially important when catering to novice programmers, including a large fraction of the biologists generating the omics data these software tools are designed to analyze. However, making it easier for beginners makes it easier for everyone, which increases the chances of the software being implemented.

The guidelines described in the following subchapter are intended to complement widely accepted best practices in software engineering, such as code quality control, testing, debugging, version control, formatting, and documentation, including comments and meaningful commit messages.

All software described in this thesis, though varying in purpose, followed these guidelines and was rapidly adopted by the bioinformatics community. Where applicable, I also adhered to these guidelines when releasing auxiliary code and workflows used for downstream analyses (Chapters 3-5), which resulted in the added benefit of maximizing reproducibility.

References

1. Katz, K. *et al.* The Sequence Read Archive: A decade more of explosive growth. *Nucleic Acids Res.* **50**, D387–D390 (2022).
2. Zappia, L., Phipson, B. & Oshlack, A. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLoS Comput. Biol.* **14**, (2018).
3. Dobin, A. *et al.* STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).
4. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
5. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32**, 381–386 (2014).
6. Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* **14**, 417–419 (2017).
7. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **34**, 525–527 (2016).
8. Dreyfuss, E. Want to Make It as a Biologist? Better Learn to Code. *WIRED* (2017).
9. Gammie, A., Lorsch, J. & Singh, S. Catalyzing the Modernization of Graduate Education. *NIGMS Feedback Loop Blog – National Institute of General Medical Sciences* (2015).
10. Luebbert, L. & Pachter, L. Efficient querying of genomic reference databases with gget. *Bioinformatics* **39**, 4–6 (2023).

Name	Platform	Citations	Website/ Manual	Manual type	Tutorials/ Vignettes
STAR	C/C++	32,005	Link	Separate website	Yes
Seurat	R	29,316	Link	Separate website	Yes
Monocle	R	7,912	Link	Bioconductor standard	Yes
salmon	C++	7,030	Link	Separate website	Yes
kallisto	C/C++	7,011	Link	Separate website	Yes
Scanpy	Python	4,481	Link	Separate website	Yes
CellRanger	Python/R	4,084	Link	Separate website	Yes
inferCNV	R	3,329	Link	GitHub Wiki	Yes
SCENIC	R/Python	3,328	Link	Separate website	Yes
Harmony	R/C++	3,241	Link	Separate website	Yes
CellPhoneDB	Python	3,103	Link	Separate website	Yes
AUCell	R	2,878	Link	Separate website	Yes
BackSPIN	Python	2,501	Link	GitHub README	Yes
velocyto	Python/R	2,497	Link	Separate website	Yes
scran	R	2,329	Link	Bioconductor standard	Yes
SingleR	R	2,129	Link	Bioconductor standard	Yes
scvi-tools	Python	2,086	Link	Separate website	Yes
CellChat	R/C++	1,948	Link	GitHub README	Yes
MAST	R	1,918	Link	Bioconductor standard	Yes
Rsubread	R	1,616	Link	Bioconductor standard	Yes
DoubletFinder	R	1,533	Link	GitHub README	Yes
batchelor	R	1,513	Link	Bioconductor standard	Yes
slingshot	R	1,470	Link	Bioconductor standard	Yes
scVelo	Python	1,454	Link	Separate website	Yes
SCDE	R	1,412	Link	Separate website	Yes
MiXCR	Java/Kotlin	1,371	Link	Separate website	Yes
UMI-tools	Python	1,253	Link	Separate website	Yes
Scrublet	Python	1,214	Link	GitHub README	Yes
Scater	R	1,212	Link	Bioconductor standard	Yes
scuttle	R/C++	1,212	Link	Bioconductor standard	Yes
MAGIC	Python/R/MATLAB	1,147	Link	GitHub README	Yes
SC3	R	1,128	Link	Bioconductor standard	Yes
MIMOSCA	Python	1,090	Link	GitHub README	Yes
dynverse	R	1,036	Link	Separate website	Yes
bseqsc	R	1,007	Link (broken)	Separate website	Yes

Table 1.1 The 35 most highly cited (>1000 citations) software tools for the analysis of scRNA-seq data published between September 2016 and December 2022 according to the scRNA-tools database (https://www.scrna-tools.org/).

INTRODUCTION – PART II

Guidelines for User-Friendly Omics Software

The below is a checklist of simple yet effective guidelines for user-friendly omics software curated based on the analyses performed in part I of the introduction:

- Devise a memorable and distinctive name for the tool that is compatible with Google and GitHub searches (good example: kallisto, poor example: bio)
    - Tip: For increased searchability on GitHub, include GitHub keywords and a short "About" description
- If possible, the computational resources (e.g., memory and disk space) to run the software should not exceed those of a standard laptop
    - If this is not feasible, consider supplying a simplified version of the software (for example, the widely used *gget alphafold* module (see Chapter 2) allows users to run a simplified version of AlphaFold2 without requiring 3 TB of disk space and a modern NVIDIA GPU)
- Provide users with a functional installation (including package dependencies) using a single line of code, ideally through the widely used package managers PyPI, Bioconda, and/or Bioconductor (e.g., "pip install gget")
- Keep package dependencies to a minimum (ideally limited to standard libraries or widely used third-party libraries, e.g., numpy), and specify dependency versions if necessary (avoid this or provide version ranges to avoid package version conflicts)
- Provide clear and coherent documentation through various media:
    - Provide function descriptions in Python/R/etc.
    - Add help (-h / --help) methods to shell scripts
    - Write a GitHub README that includes the documentation or links to the documentation
    - Write extensive documentation that includes installation instructions and ideally a "quick start" or "getting started" guide, and describe the function and the data type (e.g., int) of each argument with input examples

  - Write tutorials spanning different use cases of the software, ideally in the form of immediately executable Google Colab notebooks, and link to them in the documentation

  - To increase accessibility, use alt text for images and provide the documentation in different languages, e.g., English and Spanish

- Keep required arguments to a minimum: The simplest use case of the software tool should require as little user input as possible to simplify the process of getting started

- Write extensive unit tests and use GitHub Actions to automatically run the tests when changes to the code are committed AND in specified time intervals, e.g., once a week

  - Include a badge in the GitHub README and/or the documentation to inform users of the current test status (fail/pass)

- Maintain the software tool:

  - Maintain backward compatibility when implementing changes

  - Document the changes for each new release in the software documentation

  - Update the software when the unit tests break (Tip: Set up E-mail notifications through GitHub for failing unit tests)

  - Respond to GitHub issues raised by users (I do not recommend the use of bots that automatically close "stale" GitHub issues)

  - Update dependencies to the most recent versions as updates are released

- Facilitate and promote user feedback and contribution:

  - Set up GitHub issue templates to facilitate communication between users and software developers

  - Include contributing guidelines in the software documentation, including a detailed checklist for contributors (examples: https://docs.github.com/en/communities/setting-up-your-project-for-healthy-contributions/setting-guidelines-for-repository-contributors, https://pachterlab.github.io/gget/en/contributing.html)

Chapter 2

SOFTWARE FOR BIOLOGISTS BY BIOLOGISTS - PART I

Efficient Querying of Genomic Reference Databases with *gget*

Preamble

This chapter describes the development of the software program *gget*, which, since its release in May 2022, has been downloaded 97,000 times. *gget* is a collection of separate, but interoperable modules and has grown to 16 modules to date, including several modules contributed by the *gget* community. Beyond tackling endemic problems faced by the bioinformatics community accurately and efficiently, the success of *gget* can be attributed to the factors and guidelines described in the introduction. The rationale behind *gget* as described in the summary and introduction below is magnified when working with data from non-model organisms, as further described in Chapter 4 Part I.

Laura Luebbert, Lior Pachter (2023). Efficient querying of genomic reference databases with *gget*. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btac836

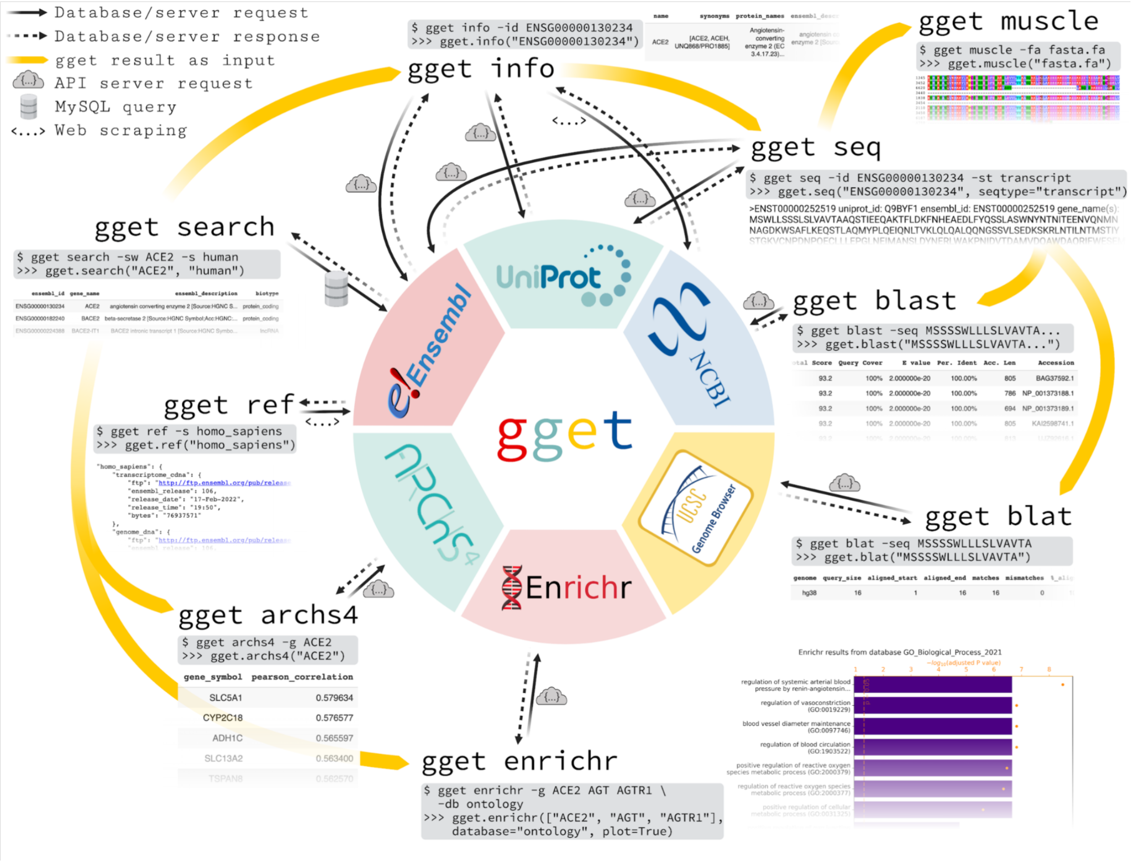
Summary

A recurring challenge in interpreting single-cell RNA-seq data is the assessment of results in the context of existing genomic databases. Currently, there is no tool implementing automated, easy programmatic access to information stored in a diverse collection of large, public genomic databases. *gget* is a free and open-source command-line tool and Python package that enables efficient querying of genomic databases. *gget* consists of a collection of separate but interoperable modules, each designed to facilitate one type of database querying required for single-cell RNA-seq data analysis in a single line of code. The manual and source code are available at https://github.com/pachterlab/gget.

Introduction

The increasingly common use of single-cell RNA-seq to provide transcriptomic characterization of cells is dependent on quick and easy access to reference information stored in large genomic databases such as Ensembl, NCBI, and UniProt (Cunningham *et al.*, 2022; NCBI Resource Coordinators, 2013; UniProt Consortium, 2021). A majority of researchers currently access genomic databases to annotate and functionally characterize putative marker genes through web access (Stalker *et al.*, 2004; Birney *et al.*, 2004). This process is time-consuming and error-prone, as it requires manually copying and pasting data, such as gene IDs.

To facilitate and automate functional annotation for single-cell RNA-seq analyses, we developed *gget:* a free and open-source software package that rapidly queries information stored in several large, public databases directly from a command line or Python environment. *gget* consists of a collection of tools designed to perform the database querying required for single-cell RNA-seq data analysis in a single line of code. In addition


Database/server request
Database/server response
gget result as input
{...} API server request
MySQL query
<...> Web scraping
gget info
gget muscle
gget seq
gget search
gget blast
gget ref
gget blat
gget archs4
gget enrichr
UniProt
Ensembl
NCBI
ARCHS4
Enrichr
UCSC Genome Browser
gget
Enrichr results from database GO_Biological_Process_2021
$-log_{10}$(adjusted P value)
1
2
3
4
5
6
7
8
p = 0.05
regulation of systemic arterial blood pressure by renin-angiotensin...
regulation of vasoconstriction (GO:0019229)
blood vessel diameter maintenance (GO:0097746)
regulation of blood circulation (GO:1903522)
positive regulation of reactive oxygen species metabolic process (GO:2000379)
regulation of reactive oxygen species metabolic process (GO:2000377)
positive regulation of cellular metabolic process (GO:0031325)

```
$ gget info -id ENSG00000130234
>>> gget.info("ENSG00000130234")
```

```
name    synonyms                   protein_names                 ensembl_descr
ACE2    [ACE2, ACEH,               Angiotensin-converting        angiotensin co
        UNQ868/PRO1885]            enzyme 2 (EC 3.4.17.23)...    enzyme 2 [Sourc
```

```
$ gget muscle -fa fasta.fa
>>> gget.muscle("fasta.fa")
```

```
$ gget seq -id ENSG00000130234 -st transcript
>>> gget.seq("ENSG00000130234", seqtype="transcript")
```

```
>ENST00000252519 uniprot_id: Q9BYF1 ensembl_id: ENST00000252519 gene_name(s):
MSWLLSSSLSLVAVTAAQSTIEEQAKTFLDKFNHEAEDLFYQSSLASWNYNTNITEENVQNMN
NAGDKWSAFLKEQSTLAQMYPLQEIQNLTVKLQLQALQQNGSSVLSEDKSKRLNTILNTMSTIY
STGKVCNPDNPQECLLLEPGLNEIMANSLDYNERLWAKPNIDVTDAMVDQAWDAQRIFWESEM
```

```
$ gget search -sw ACE2 -s human
>>> gget.search("ACE2", "human")
```

```
ensembl_id        gene_name  ensembl_description                                  biotype
ENSG00000130234   ACE2       angiotensin converting enzyme 2 [Source:HGNC S...    protein_coding
ENSG00000182240   BACE2      beta-secretase 2 [Source:HGNC Symbol;Acc:HGNC:...    protein_coding
ENSG00000224388   BACE2-IT1  BACE2 intronic transcript 1 [Source:HGNC Symbo...    lncRNA
```

```
$ gget blast -seq MSSSSWLLLSLVAVTA...
>>> gget.blast("MSSSSWLLLSLVAVTA...")
```

```
otal Score  Query Cover  E value       Per. Ident  Acc. Len  Accession
      93.2         100%  2.000000e-20     100.00%       805  BAG37592.1
      93.2         100%  2.000000e-20     100.00%       786  NP_001373188.1
      93.2         100%  2.000000e-20     100.00%       694  NP_001373189.1
      93.2         100%  2.000000e-20     100.00%       805  KAI2598741.1
      93.2         100%  2.000000e-20     100.00%       813  UJZ92616.1
```

```
$ gget ref -s homo_sapiens
>>> gget.ref("homo_sapiens")
```

```
"homo_sapiens": {
    "transcriptome_cdna": {
        "ftp": "http://ftp.ensembl.org/pub/release
        "ensembl_release": 106,
        "release_date": "17-Feb-2022",
        "release_time": "19:50",
        "bytes": "76937571"
    },
    "genome_dna": {
        "ftp": "http://ftp.ensembl.org/pub/release
        "ensembl_release": 106,
```

```
$ gget blat -seq MSSSSWLLLSLVAVTA
>>> gget.blat("MSSSSWLLLSLVAVTA")
```

```
genome  query_size  aligned_start  aligned_end  matches  mismatches  %_alig
hg38            16              1           16       16           0
```

```
$ gget archs4 -g ACE2
>>> gget.archs4("ACE2")
```

```
gene_symbol  pearson_correlation
     SLC5A1             0.579634
    CYP2C18             0.576577
      ADH1C             0.565597
    SLC13A2             0.563400
     TSPAN8             0.562570
```

```
$ gget enrichr -g ACE2 AGT AGTR1 \
  -db ontology
>>> gget.enrichr(["ACE2", "AGT", "AGTR1"],
    database="ontology", plot=True)
```

Figure 2.1 Overview of the nine *gget* tools and the public databases they access. One simple command line ($) example and its Python (>>>) equivalent are shown for each tool with the corresponding output.

to providing access to genomic databases, *gget* can also leverage sequence analysis tools, such as BLAST (Altschul *et al.*, 1990, 1997), thus simplifying complex annotation workflows.

While there are some web-based Application Programming Interface (API) data mining systems, such as BioMart (Durinck *et al.*, 2005; Kasprzyk *et al.*, 2004), we identified several limitations in such tools, including limits to query types and to utilizing databases in tandem. Moreover, large-scale single-cell RNA-seq analysis is better served by command line or packaged APIs that can fetch data directly into programming environments.

The *gget* modules combine MySQL (Oracle Corporation, 1995), API, and web data extraction queries to rapidly and reliably request comprehensive information from different databases (Figure 2.1). This approach allows *gget* to perform tasks unsupported by existing tools built around standard API queries (de Ruiter, 2016). For instance, searching for genes and transcripts using free-form search terms. Each *gget* tool requires minimal arguments, provides clear output, and operates from both the command line and Python environments, such as JupyterLab, maximizing ease of use and accommodating novice programmers.

Description

gget consists of nine tools:

- *gget ref:* Fetch File Transfer Protocols (FTPs) and metadata for reference genomes or annotations from Ensembl by species.
- *gget search:* Fetch genes or transcripts from Ensembl using free-form search terms.
- *gget info*: Fetch extensive gene or transcript metadata from Ensembl, UniProt, and NCBI by Ensembl ID.
- *gget seq:* Fetch nucleotide or amino acid sequences of genes or transcripts from Ensembl or UniProt by Ensembl ID.
- *gget blast:* BLAST (Altschul *et al.*, 1990, 1997) a nucleotide or amino acid sequence to any BLAST database.
- *gget blat:* Find the genomic location of a nucleotide or amino acid sequence using BLAT (James Kent, 2002).
- *gget muscle:* Align multiple nucleotide or amino acid sequences to each other using the Muscle5 algorithm (Edgar, 2021).
- *gget enrichr:* Perform an enrichment analysis on a list of genes using Enrichr (Chen *et al.*, 2013; Xie *et al.*, 2021; Kuleshov *et al.*, 2016) and an extensive collection of gene set libraries, including KEGG (Kanehisa and Goto, 2000; Kanehisa, 2019; Kanehisa *et al.*, 2021) and Gene Ontology (Ashburner *et al.*, 2000; Gene Ontology Consortium, 2021).
- *gget archs4:* Find the most correlated genes to a gene of interest or find the gene's tissue expression atlas using ARCHS4 (Lachmann *et al.*, 2018).

Each *gget* tool accesses data stored in one or several public databases, as depicted in Figure 2.1. *gget* fetches the requested data in real-time, guaranteeing that each query will return the latest information. One exception is *gget muscle*, which locally compiles the Muscle5 algorithm (Edgar, 2021) and therefore does not require an internet connection.

gget info combines information from Ensembl, NCBI, and UniProt (Cunningham *et al.*, 2022; NCBI Resource Coordinators, 2013; UniProt Consortium, 2021) to provide the user with a comprehensive executive summary of the available information about a gene or transcript. This also enables users to assert whether data from different sources are consistent.

By accessing the NCBI server (NCBI Resource Coordinators, 2013) through HTTP requests, *gget blast* does not require the download of a reference BLAST database, as is the case with existing BLAST tools (Buchfink *et al.*, 2021; Camacho *et al.*, 2009). The whole self-contained *gget* package is approximately 3 MB after installation.

The package dependencies were carefully chosen and kept to a minimum. *gget* depends on the HTML parser *beautifulsoup4* (Richardson, 2022), the Python MySQL-connector (Oracle, 2022), and the HTTP library *requests* (Reitz, 2022). All of these are well-established packages for server interaction in Python. *gget* has been tested on Linux/Unix, Mac OS (Darwin), and Windows.

Usage and documentation

gget can be installed from the command line by running ‘pip install gget’. Figure 2.1 depicts one use case for each *gget* tool with the corresponding output.

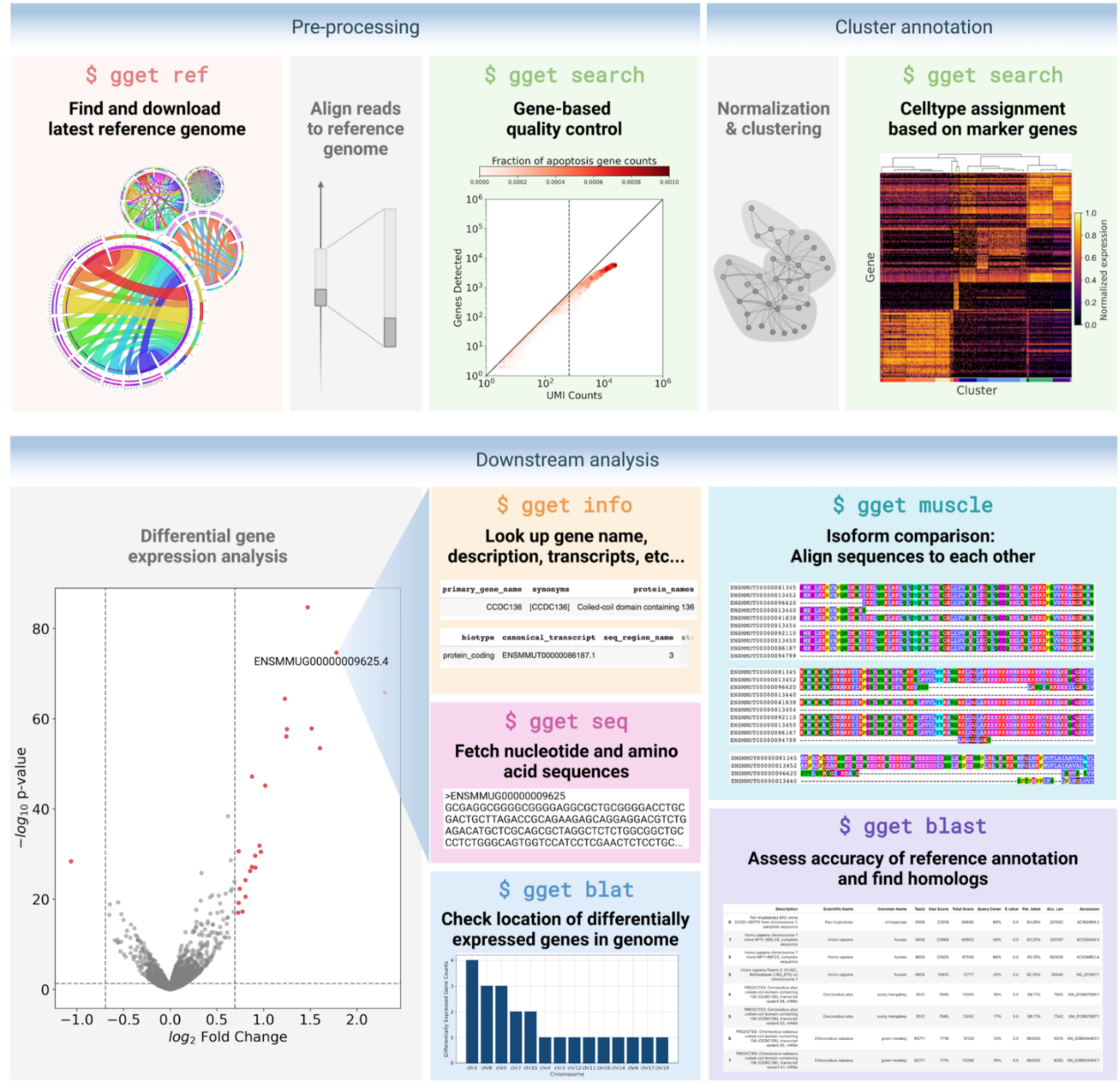
Each *gget* tool features an extensive manual available as function documentation in a Python environment or as standard output using the help flag [-h] in the command line. The complete manual with examples can be viewed in the *gget* repository, available at https://github.com/pachterlab/gget. A separate *gget examples* repository is accessible at https://github.com/pachterlab/gget_examples and includes exemplary workflows immediately executable in Google Colaboratory (Bisong, 2019).

Discussion

Our open-source Python and command-line program *gget* enables efficient and easy programmatic access to information stored in a diverse collection of large, public genomic databases. The *gget* modules were motivated by experience with tedious single-cell RNA-seq data analysis tasks (Supplementary Figure 2.1), however, we anticipate their utility for a wide range of bioinformatics tasks.

Acknowledgments

We thank Kyung Hoi (Joseph) Min for advice on the command-line interface, Matteo Guareschi for advice on Windows operability, and A. Sina Booeshaghi, Kristján Eldjárn Hjörleifsson, and Ángel Gálvez-Merchán for insightful discussions about *gget*. Cartoons in Figure 2.1 and Supplementary Figure 2.1 were created with BioRender.com. Thanks to the wonderful staff at Dash Coffee Bar in Pasadena, who occasionally gave LL free banana bread to sustain this work.


Pre-processing
Cluster annotation
$ gget ref
Find and download latest reference genome
Align reads to reference genome
$ gget search
Gene-based quality control
Fraction of apoptosis gene counts
0.0000
0.0002
0.0004
0.0006
0.0008
0.0010
Genes Detected
UMI Counts
Normalization & clustering
$ gget search
Celltype assignment based on marker genes
Gene
Cluster
Normalized expression
1.0
0.8
0.6
0.4
0.2
0.0
Downstream analysis
Differential gene expression analysis
ENSMMUG00000009625.4
$-log_{10}$ p-value
log_2 Fold Change
80
60
40
20
0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
$ gget info
Look up gene name, description, transcripts, etc...
$ gget seq
Fetch nucleotide and amino acid sequences
$ gget blat
Check location of differentially expressed genes in genome
Differentially Expressed Gene Counts
Chromosome
$ gget muscle
Isoform comparison: Align sequences to each other
$ gget blast
Assess accuracy of reference annotation and find homologs

primary_gene_name	synonyms	protein_names
CCDC136	[CCDC136]	Coiled-coil domain containing 136

biotype	canonical_transcript	seq_region_name
protein_coding	ENSMMUT00000086187.1	3

```
>ENSMMUG00000009625
GCGAGGCGGGGCGGGGAGGCGCTGCGGGGACCTGC
GACTGCTTAGACCGCAGAAGAGCAGGAGGACGTCTG
AGACATGCTCGCAGCGCTAGGCTCTCTGGCGGCTGC
CCTCTGGGCAGTGGTCCATCCTCGAACTCTCCTGC...
```

Supplementary Figure 2.1 *gget* performs the database querying underlying a standard single-cell RNA-seq data analysis workflow. The workflow and all of the figures are reproducible starting with raw reads using immediately executable Google Colaboratory notebooks that can be run for free and are accessible at https://github.com/pachterlab/gget_examples/tree/main/scRNAseq_workflow.

References

Altschul, S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul, S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ashburner, M. *et al.* (2000) Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

Birney, E. *et al.* (2004) An overview of Ensembl. *Genome Res.*, **14**, 925–928.

Bisong, E. (2019) Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 59–64.

Buchfink, B. *et al.* (2021) Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods*, **18**, 366–368.

Camacho, C. *et al.* (2009) BLAST+: Architecture and applications. *BMC Bioinformatics*, **10**, 421.

Chen, E.Y. *et al.* (2013) Enrichr: Interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics*, **14**, 128.

Cunningham, F. *et al.* (2022) Ensembl 2022. *Nucleic Acids Res.*, **50**, D988–D995.

Durinck, S. *et al.* (2005) BioMart and Bioconductor: A powerful link between biological databases and microarray data analysis. *Bioinformatics*, **21**, 3439–3440.

Edgar, R.C. (2021) MUSCLE v5 enables improved estimates of phylogenetic tree confidence by ensemble bootstrapping. *bioRxiv*.

Gene Ontology Consortium (2021) The Gene Ontology resource: Enriching a GOld mine. *Nucleic Acids Res.*, **49**, D325–D334.

James Kent, W. (2002) BLAT—The BLAST-Like Alignment Tool. *Genome Res.*, **12**, 656–664.

Kanehisa, M. *et al.* (2021) KEGG: Integrating viruses and cellular organisms. *Nucleic Acids Res.*, **49**, D545–D551.

Kanehisa, M. (2019) Toward understanding the origin and evolution of cellular organisms. *Protein Sci.*, **28**, 1947–1951.

Kanehisa, M. and Goto,S. (2000) KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.

Kasprzyk, A. *et al.* (2004) EnsMart: A generic system for fast and flexible access to biological data. *Genome Res.*, **14**, 160–169.

Kuleshov, M.V. *et al.* (2016) Enrichr: A comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res.*, **44**, W90–7.

Lachmann, A. *et al.* (2018) Massive mining of publicly available RNA-seq data from human and mouse. *Nat. Commun.*, **9**, 1366.

NCBI Resource Coordinators (2013) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **41**, D8–D20.

Oracle (2022) mysql-connector-python 8.0.29.

Oracle Corporation (1995) MySQL https://www.mysql.com/.

Reitz, K. (2022) requests 2.27.1.

Richardson, L. (2022) beautifulsoup4 4.11.1.

de Ruiter, J. (2016) PyBiomart 0.2.0 https://jrderuiter.github.io/pybiomart/.

Stalker, J. *et al.* (2004) The Ensembl Web site: Mechanics of a genome browser. *Genome Res.*, **14**, 951–955.

UniProt Consortium (2021) UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Res.*, **49**, D480–D489.

Xie, Z. *et al.* (2021) Gene Set Knowledge Discovery with Enrichr. *Curr. Protoc.*, **1**, e90.

SOFTWARE FOR BIOLOGISTS BY BIOLOGISTS - PART II

Fast and Scalable Querying of Eukaryotic Linear Motifs with *gget elm*

Preamble

This subchapter describes the development of *gget elm*, a recently published tool for high-throughput identification of protein-protein interaction motifs in amino acid sequences. The *gget elm* module exemplifies how the *gget* project provides a platform and backbone for the rapid development of novel modules that solve widely faced challenges in computational biology spanning various fields, in this case interactomics, proteomics, and molecular cell biology.

Laura Luebbert, Chi Hoang, Manjeet Kumar, Lior Pachter (2023). Fast and scalable querying of eukaryotic linear motifs with *gget elm*. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btae095

Summary

Eukaryotic linear motifs (ELMs), or Short Linear Motifs (SLiMs), are protein interaction modules that play an essential role in cellular processes and signaling networks and are often involved in diseases like cancer. The ELM database is a collection of manually curated motif knowledge from scientific papers. It has become a crucial resource for investigating motif biology and recognizing candidate ELMs in novel amino acid sequences. Users can search amino acid sequences or UniProt Accessions on the ELM resource web interface. However, as with many web services, there are limitations in the swift processing of large-scale queries through the ELM web interface or API calls, and, therefore, integration into protein function analysis pipelines is limited.

To allow swift, large-scale motif analyses on protein sequences using ELMs curated in the ELM database, we have extended the *gget* suite of Python and command line tools with a new module, *gget elm*, which does not rely on the ELM server for efficiently finding candidate ELMs in user-submitted amino acid sequences and UniProt Accessions. *gget elm* increases accessibility to the information stored in the ELM database and allows scalable searches for motif-mediated interaction sites in the amino acid sequences.

The manual and source code are available at https://github.com/pachterlab/gget.

Introduction

Eukaryotic linear motifs (ELMs), also known as Short Linear Motifs (SLiMs), are short stretches of contiguous amino acids, typically 3 to 15 residues in length, encoding protein-protein interaction sites. They are mainly located in the intrinsically disordered regions (IDRs) of proteins and are typically found to be highly conserved in orthologous proteins. These modules can encode multiple functionalities, which include modification, degradation, docking, targeting, and binding sites for protein domains. As such, ELM-mediated interactions play an essential role in cellular processes and signaling networks, including the regulation of homeostasis, apoptosis, and differentiation (Van Roey *et al.*, 2014; Davey *et al.*, 2012). Pathogens like SARS-CoV-2 mimic ELMs to gain entry into the cell (Kruse *et al.*, 2021; Mészáros *et al.*, 2021), and mutations in sequences containing ELMs contribute to diseases like cancer (Uyar *et al.*, 2014; Mészáros *et al.*, 2017). As a result, ELM-mediated protein interactions are potential targets for therapeutic intervention (Mészáros *et al.*, 2021; Simonetti *et al.*, 2023; Fasano *et al.*, 2022).

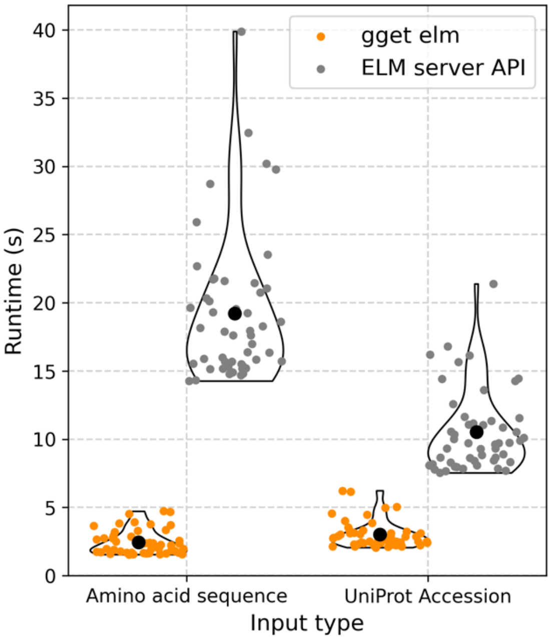


Figure 2.2 Runtime comparison for 50 amino acid sequences and 50 UniProt Accessions submitted to *gget elm* and the ELM server API. For the ELM server API, a 3-minute wait time was observed between each request to comply with the server rules. These wait times were not taken into account when measuring the runtimes. The black dot denotes the mean. The code used to generate this figure can be found here: http://tinyurl.com/bdc6mhm3.

The ELM resource has two main components: an exploratory candidate motif search web interface and a database with manually curated linear motif knowledge, including information on binding partners and recognition features along with associated biological context. The database information is derived from the scientific literature by expert ELM curators who analyze motif-containing sequences to capture key insights, such as the residues involved in the interaction, their evolutionary conservation, local sequence context in flanking regions, features of the binding site on the interacting partner, and other motif-related insights. In addition, the curation process captures relevant information on the contextual knowledge, which includes cellular function, location, and taxonomic distribution of motif-containing proteins. Since the database was first created (Puntervoll *et al.*, 2003; Dinkel *et al.*, 2011), it has been continuously updated and has been widely used for both biomedical studies as well as interactomics, proteomics, and molecular research studies (Kumar *et al.*, 2020, 2022; Gouw *et al.*, 2018; Dinkel *et al.*, 2015; Carberry, 2008; Kumar *et al.*, 2023, Benz *et al.*, 2022; Gogl *et al.*, 2022; Zhang *et al.*, 2012; Reys and Labesse, 2022). Users can search amino acid sequences or UniProt Accessions on the ELM database web interface (http://elm.eu.org/) or by submitting an API request through the ELM server. However, these methods have processing limitations when performing large-scale queries, and many requests being submitted simultaneously can lead to server overload and extended wait times.

To expedite the investigation of ELMs, we have extended the *gget* suite of Python and command line tools (Luebbert and Pachter, 2022) with a new module which efficiently finds ELMs in user-submitted amino acid sequences or UniProt Accessions: *gget elm*. *gget elm* increases accessibility to the information stored in the ELM database and allows scalable searches for ELMs in amino acid sequences. The command line interface and

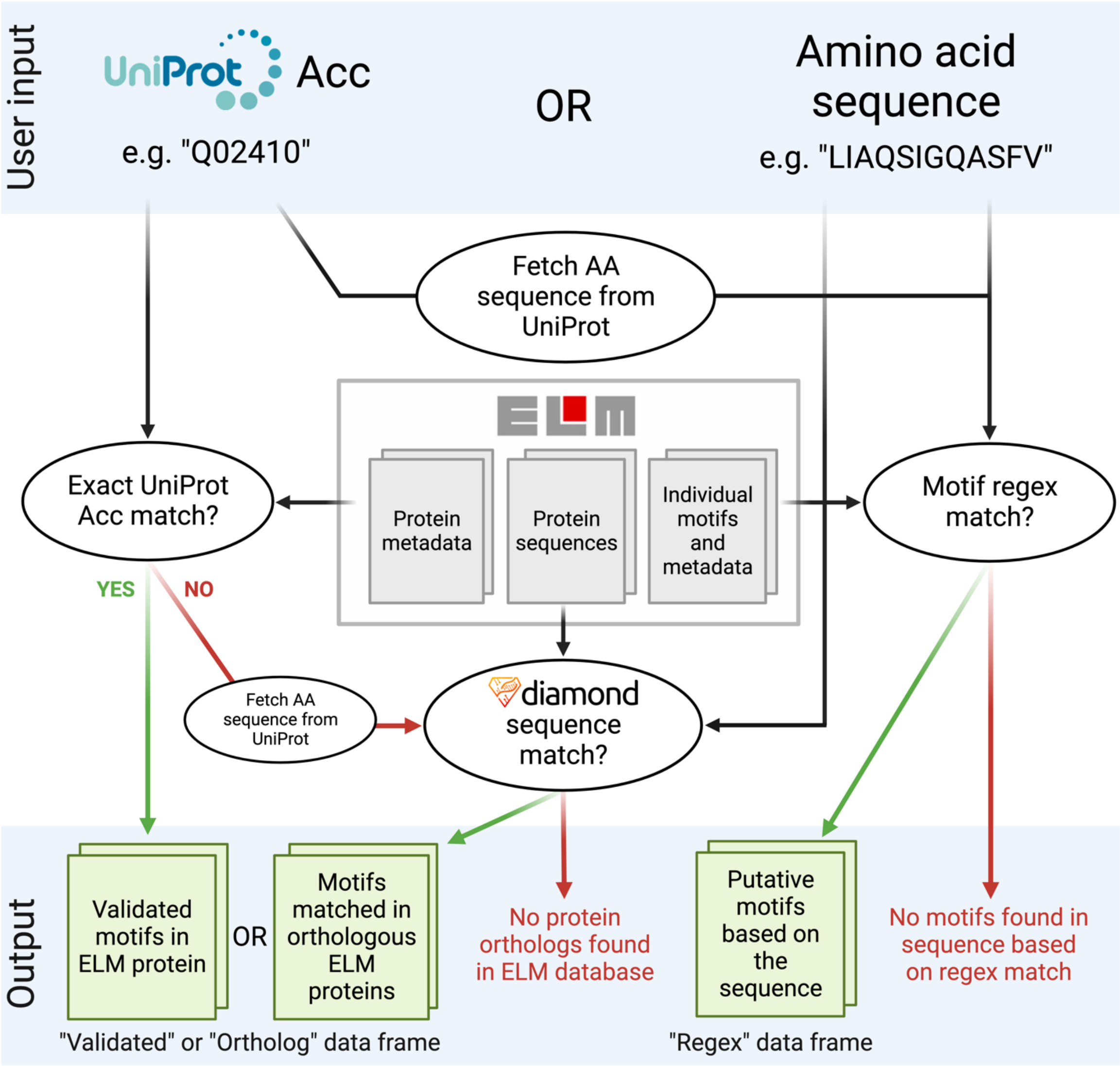

User input
UniProt Acc
e.g. "Q02410"
OR
Amino acid sequence
e.g. "LIAQSIGQASFV"
Fetch AA sequence from UniProt
ELM
Protein metadata
Protein sequences
Individual motifs and metadata
Exact UniProt Acc match?
Motif regex match?
YES
NO
Fetch AA sequence from UniProt
diamond sequence match?
Output
Validated motifs in ELM protein
OR
Motifs matched in orthologous ELM proteins
No protein orthologs found in ELM database
Putative motifs based on the sequence
No motifs found in sequence based on regex match
"Validated" or "Ortholog" data frame
"Regex" data frame

Figure 2.3 Schematic overview of the *gget elm* back-end.

optional JSON formatted output allow swift integration into existing protein analysis workflows.

Description

Users can submit an amino acid sequence or a UniProt Accession to *gget elm*. *gget elm* captures both homology-based matches corresponding to curated motifs in orthologous proteins in the ELM database and POSIX regular expression (regex) matches corresponding to candidate motifs in the provided sequence. Hence, *gget elm* returns two separate data frames (or JSON formatted dictionaries for use from the command line) containing the respective motif matches and extensive information about each motif. Figure 2.3 provides an overview of the *gget elm* back-end.

After installing *gget* ($ pip install gget), the user downloads the ELM database reference information using a specialized module, *gget setup*, with the command $ gget setup elm. This command may be repeated at any time to update the local copy of the ELM database, which currently requires a total of 3 MB of disk space. The files are saved in the *gget* installation directory. If the user submits a UniProt Accession to *gget elm* and the protein is not present in the ELM database, its amino acid sequence is fetched from UniProt (UniProt Consortium, 2021). Using the DIAMOND alignment algorithm (Buchfink *et al.*, 2021), the sequence is compared to the motif-containing proteins in the ELM database. *gget elm* returns all motifs associated with orthologous proteins, including information about each orthologous protein, and extensive details on each motif. *gget elm* also returns alignment scores for each DIAMOND hit, including identity and coverage percentages and boolean output on whether the orthologous motif is contained within the overlapping region between the query and subject sequence. To compute the regex data frame, *gget elm* considers all regex expressions from the ELM database and scans them against the provided amino acid sequence to report all matches. The data from the ELM database is combined to return relevant information about each matched interaction motif, including motif description, type, sequence, location in the ortholog and query sequence, and host taxonomy, for both data frames. How different types of user input traverse the *gget elm* back-end is explored in this Google Colab notebook: https://tinyurl.com/4bd5h8hr.

gget elm builds on existing *gget* modules, such as *gget seq* to fetch amino acid sequences from UniProt, and a new module developed in parallel with *gget elm*: *gget diamond*, which aligns sequences using the DIAMOND algorithm (Buchfink *et al.*, 2021) and can be used independently from *gget elm*.

While *gget elm* results are similar to results obtained through the ELM web interface, they may not be identical due to differences underlying the computations. For example, *gget elm* uses DIAMOND for fast and sensitive local alignment of the amino acid sequences, whereas the ELM web interface has its own suite of back-end tools and deliberately limits the number of proteins in the output to be manageable for the web server (Chica *et al.*, 2008). In a comparison between the 'regex' data frame returned by *gget elm* and the results obtained through the ELM server API for 50 amino acid sequences and 50 UniProt Accessions, *gget elm* returned results 8x faster for amino acid sequences and 3.5x faster for UniProt Accessions on average (Figure 2.2). For the ELM server API, runtimes are further increased significantly by a mandatory 1-minute wait time between amino acid sequence requests, and a 3-minute wait time between UniProt Accession requests to comply with the server usage recommendations and avoid 429 errors. The results returned by both methods matched 100% across all tested amino acid sequences and UniProt Accessions. The code to reproduce this analysis can be found here: http://tinyurl.com/bdc6mhm3.

Usage and Documentation

Akin to all modules contained within *gget* (Luebbert and Pachter, 2022)*, gget elm* features an extensive manual available as function documentation in a Python environment or as standard output using the help flag [-h] in the command line. The accuracy of the returned results is maintained through extensive unit tests, which automatically run on a bi-weekly basis. The complete manual with examples can be viewed on the *gget* website in English (https://pachterlab.github.io/gget/en/elm) and in Spanish (https://pachterlab.github.io/gget/es/elm).

gget can be installed from PyPI using the command line with the following command:

```
$ pip install gget
```

Alternatively, *gget* can be installed using Anaconda:

```
$ conda install -c bioconda gget
```

Example *gget elm* commands to find ELMs in a protein from its amino acid sequence or UniProt Accession look as follows:

Command line (JSON formatted results are saved in a folder named ‘results’):

```
$ gget setup elm                # Downloads/updates local ELM database
$ gget elm -o results LIAQSIGQASFV
$ gget elm -o results --uniprot Q02410
```

Python (two data frames are returned):

```
>>> gget.setup(“elm”)           # Downloads/updates local ELM database
>>> ortholog_df, regex_df = gget.elm(“LIAQSIGQASFV”)
>>> ortholog_df, regex_df = gget.elm(“Q02410”, uniprot=True)
```

The [--threads][-t] (Python: “threads”) argument can be used to multithread the sequence alignment for increased speed for large-scale computations. The following tutorial demonstrates how *gget elm* can be combined with the IUPred3 API (Erdős *et al.,* 2021) to filter putative ELMs located within intrinsically disordered regions and thereby limiting false positive matches: http://tinyurl.com/mw5s5yf3.

Proof of concept: *gget elm* reports the loss of a protein interaction motif involved in DNA repair in a carcinogenic BRCA2 mutation

BRCA2 (BReast CAncer gene 2) plays an essential role in DNA repair through homologous recombination, and heterozygous germline defects in BRCA2 increase the risk of breast cancer. The promotion of homologous recombination by BRCA2 requires its association with the partner and localizer of BRCA2 (PALB2) (Hanenberg and Andreassen, 2018). This important protein-protein interaction occurs at the site of a linear motif (ELM: LIG_PALB2_WD40_1, regex: [....WF..L]), which can be recognized by *gget elm*. We analyzed the wildtype BRCA2 sequence and a mutant BRCA2 sequence with a single amino acid substitution (W31C), previously described as carcinogenic due to a loss of interaction with PALB2 (Oliver *et al.*, 2009). *gget elm* accurately reports the loss of the PALB2 interaction motif in the mutant sequence compared to the wildtype sequence: https://tinyurl.com/yc5r2b5m.

Discussion

We have shown that *gget elm* facilitates scalable querying of the ELM database via local queries, and its use via the command line makes it easy to integrate into scripted workflows. While this feature should extend the usability of the ELM database, there are limitations while performing motif searches using the ELM database web interface or *gget elm.* A common problem encountered is that short and degenerate ELMs inevitably lead to false positive matches. Accuracy can be improved by filtering the results using the additional contextual information, which is also returned by *gget elm*, including description, structural features, and host taxonomy. Furthermore, combining motif results with structural and alignment information can provide information about the functional availability of the interaction site (Lee *et al.*, 2023). The 3D structure of a protein can be predicted from its amino acid sequence *de novo* using algorithms like AlphaFold2 (Jumper *et al.*, 2021) and compared to experimentally derived crystal structures of orthologs deposited on the PDB (Berman *et al.*, 2000). The *gget* suite of tools contains a workflow to perform both of these computations, which is demonstrated here: https://tinyurl.com/yzc9ytvx.

Acknowledgments

We thank the expert curators of the ELM database for providing an excellent resource. We thank Dr. Toby Gibson for the valuable feedback on the manuscript. We also thank Candace Rypisi and the rest of the Summer Undergraduate Research Fellowships (SURF) program staff for facilitating valuable research opportunities for undergraduate students and mentorship opportunities for graduate students at Caltech. Illustrations in Figure 2.3 were created with BioRender.com.

References

Benz, C. *et al.* (2022) Proteome-scale mapping of binding sites in the unstructured regions of the human proteome. *Mol. Syst. Biol.,* **18**, e10584.

Berman, H.M. *et al.* (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.

Buchfink, B. *et al.* (2021) Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods*, **18**, 366–368.

Carberry, J.,Jr (2008) Toward a unified theory of high-energy metaphysics: Silly string theory. *Knit Forecast Int.*, **5**, 1–3.

Chica, C. *et al.* (2008) A tree-based conservation scoring method for short linear motifs in multiple alignments of protein sequences. *BMC Bioinformatics*, **9**, 229.

Davey, N.E. *et al.* (2012) Attributes of short linear motifs. *Mol. Biosyst.*, **8**, 268–281.

Dinkel, H. *et al.* (2015) ELM 2016—data update and new functionality of the eukaryotic linear motif resource. *Nucleic Acids Res.*, **44**, D294–D300.

Dinkel, H. *et al.* (2011) ELM—the database of eukaryotic linear motifs. *Nucleic Acids Res.*, **40**, D242–D251.

Erdős, G. *et al.* (2021) IUPred3: Prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation. *Nucleic Acids Res.*, **49**, W297–W303.

Fasano, C. *et al.* (2022) Short Linear Motifs in Colorectal Cancer Interactome and Tumorigenesis. *Cells*, **11**.

Gogl, G. *et al.* (2022) Quantitative fragmentomics allow affinity mapping of interactomes. *Nat. Commun.*, **13**, 5472.

Gouw, M. *et al.* (2018) The eukaryotic linear motif resource - 2018 update. *Nucleic Acids Res.*, **46**, D428–D434.

Hanenberg, H. and Andreassen,P.R. (2018) PALB2 (partner and localizer of BRCA2). *Atlas Genet. Cytogenet. Oncol. Haematol.*, **22**, 484–490.

Jumper, J. *et al.* (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583–589.

Kruse, T. *et al.* (2021) Large scale discovery of coronavirus-host factor protein interaction motifs reveals SARS-CoV-2 specific mechanisms and vulnerabilities. *Nat. Commun.*, **12**, 6761.

Kumar, M. *et al.* (2023) ELM-the Eukaryotic Linear Motif resource-2024 update. *Nucleic Acids Res.*

Kumar, M. *et al.* (2020) ELM-the Eukaryotic Linear Motif resource in 2020. *Nucleic Acids Res.*, **48**, D296–D306.

Kumar, M. *et al.* (2022) The Eukaryotic Linear Motif resource: 2022 release. *Nucleic Acids Res.*, **50**, D497–D508.

Lee, C.Y. *et al.* (2023) Systematic discovery of protein interaction interfaces using AlphaFold and experimental validation. *bioRxiv*, 2023.08.07.552219.

Luebbert, L. *et al.* (2023) Efficient querying of genomic reference databases with gget. *Bioinformatics*, **39**, btac836.

Mészáros, B. *et al.* (2017) Degrons in cancer. *Sci. Signal.*, **10**.

Mészáros, B. *et al.* (2021) Short linear motif candidates in the cell entry system used by SARS-CoV-2 and their potential therapeutic implications. *Sci. Signal.*, **14**, eabd0334.

Oliver, A.W. *et al.* (2009) Structural basis for recruitment of BRCA2 by PALB2. *EMBO Rep.*, **10**, 990–996.

Puntervoll, P. *et al.* (2003) ELM server: A new resource for investigating short functional sites in modular eukaryotic proteins. *Nucleic Acids Res.*, **31**, 3625–3630.

Reys, V. and Labesse, G. (2022) SLiMAn: An Integrative Web Server for Exploring Short Linear Motif-Mediated Interactions in Interactomes. *J. Proteome Res.*, **21**, 1654–1663.

Simonetti, L. *et al.* (2023) SLiM-binding pockets: An attractive target for broad-spectrum antivirals. *Trends Biochem. Sci.*, **48**, 420–427.

UniProt Consortium (2021) UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Res.*, **49**, D480–D489.

Uyar, B. *et al.* (2014) Proteome-wide analysis of human disease mutations in short linear motifs: neglected players in cancer? *Mol. Biosyst.*, **10**, 2626–2642.

Van Roey, K. *et al.* (2014) Short linear motifs: Ubiquitous and functionally diverse protein interaction modules directing cell regulation. *Chem. Rev.*, **114**, 6733–6778.

Zhang, Q.C. *et al.* (2012) PrePPI: A structure-informed database of protein–protein interactions. *Nucleic Acids Res.*, **41**, D828–D833.

Chapter 3

QUANTIFYING HIDDEN INFORMATION

Detection of Viral RNA at Single-cell Resolution

Preamble

Combining extensive expertise in biology and computer science allows the extraction of information from existing data that was previously not possible to quantify. In this chapter, I expanded the open-source transcriptomic data pre-processing tool kallisto to perform translated alignment of nucleotide sequences to an amino acid reference. To date, this is the only software tool capable of translated alignment while retaining single-cell resolution. I used translated alignment to identify viral RNA in bulk and single-cell RNA sequencing data based on highly conserved motifs, thereby overcoming limitations due to the lack of viral reference genomes. The single-cell resolution allowed the characterization of viral tropism and the prediction of viral presence based on host gene expression. This approach revealed novel viruses whose presence perturbed host gene expression.

Laura Luebbert, Delaney K. Sullivan, Maria Carilli, Kristján Eldjárn Hjörleifsson, Alexander Viloria Winnett, Tara Chari, Lior Pachter (2023). Efficient and accurate detection of viral sequences at single-cell resolution reveals novel viruses perturbing host gene expression. *bioRxiv*. https://doi.org/10.1101/2023.12.11.571168

Abstract

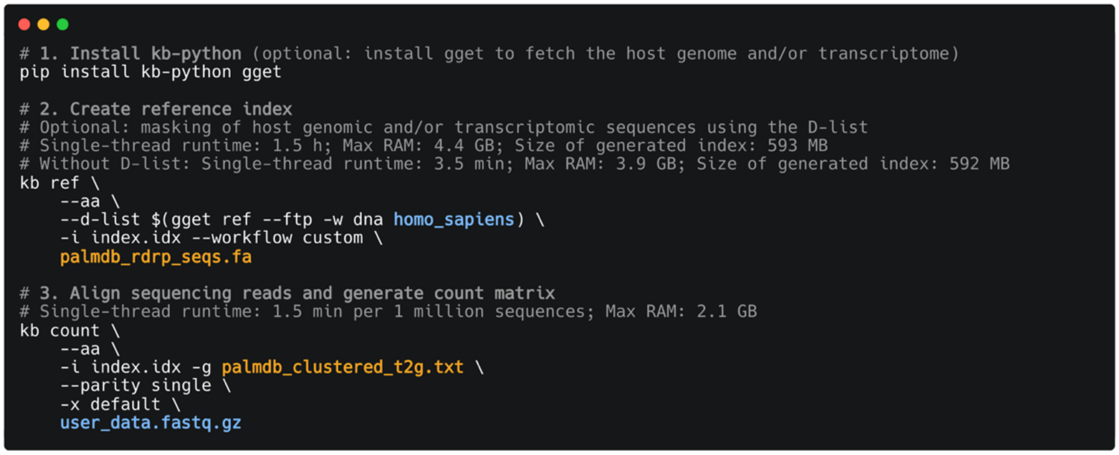
There are an estimated 300,000 mammalian viruses from which infectious diseases in humans may arise. They inhabit human tissues such as the lungs, blood, and brain and often remain undetected. Efficient and accurate detection of viral infection is vital to understanding its impact on human health and to make accurate predictions to limit adverse effects, such as future epidemics. The increasing use of high-throughput sequencing methods in research, agriculture, and healthcare provides an opportunity for the cost-effective surveillance of viral diversity and investigation of virus-disease correlation. However, existing methods for identifying viruses in sequencing data rely on and are limited to reference genomes or cannot retain single-cell resolution through cell barcode tracking. We introduce a method that accurately and rapidly detects viral sequences in bulk and single-cell transcriptomics data based on highly conserved amino acid domains, which enables the detection of RNA viruses covering over 100,000 virus species. The analysis of viral presence and host gene expression in parallel at single-cell resolution allows for the characterization of host viromes and the identification of viral tropism and host responses. We applied our method to identify putative novel viruses in rhesus macaque PBMC data that display cell type specificity and whose presence correlates with altered host gene expression.

Introduction

There are an estimated 10^{31} virions on Earth, which amounts to 10 million virions for every star in the known universe[1,2]. Viruses inhabit oceans, forests, deserts, and human tissues such as the lungs, blood, and brain. There are an estimated 300,000 mammalian viruses[3] from which infectious diseases in humans may arise[4]. However, only 261 species have been detected in humans[5]. Many of these have been implicated in complex diseases such as heart disease and cancer. Recent studies suggest that viruses also play a major, unexpected role in common neurodegenerative disorders such as Alzheimer's, Parkinson's, and multiple sclerosis[6-8]. Accurate detection of viral infections is crucial to understanding viral impact on human health.

Of the 261 known disease-causing viruses, 206 fall into the realm of *Riboviria*[5], which includes all RNA-dependent RNA polymerase (RdRP)-encoding RNA viruses and RNA-dependent DNA polymerase (RdDP)-encoding retroviruses. Amongst many others, these include Corona-, Dengue, Ebola-, Hepatitis B, influenza, Measles, Mumps, Polio-, West Nile, and Zika viruses. Most existing workflows for detecting viruses in transcriptomics data rely on the availability of pre-assembled reference genomes. Currently, NCBI RefSeq hosts 5,970 *Riboviria* reference genomes—a diminutive fraction of *Riboviria* viruses. Pioneering work by Edgar *et al.*[9] leveraged a well-conserved amino acid sub-sequence of the RdRP, called the 'palmprint', to identify RNA viruses in 5.7 million globally and ecologically diverse sequencing samples from the Sequence Read Archive (SRA). Their method's independence from pre-computed indices allowed alignment to diverged sequences, leading to the discovery of thousands of novel viruses. This effort resulted in a consensus of 296,623 unique RdRP-containing amino acid sequences, henceforth referred to as 'PalmDB'. Clustering palmprints into species-like operational taxonomic units (sOTUs) yielded 146,973 known as well as novel sOTUs[9]. Compared to the 8,694 *Riboviria* reference genomes currently available on NCBI, this translates to a more than 16x increase in the number of viruses that can be detected. The actual number of virus species that can be detected using the PalmDB is likely even higher due to RdRP sequence conservation across *Riboviria* (Extended Data Fig. 1). sOTUs serve to approximate taxonomic assignment[9,10] and allow species-level virus identification for 40,392 sequences in the PalmDB.

The increasing use of high-throughput next-generation sequencing (NGS) methods in molecular biology research, agriculture, and healthcare provides an opportunity for the cost-effective surveillance of viral diversity and the investigation of virus-disease correlations[11,12]. Specifically, single-cell genomics technologies make possible, in principle, the characterization of viruses at single-cell resolution. We expanded the RNA sequencing data preprocessing tool kallisto[13] to support the detection of viral RNA using the amino acid database PalmDB. To our knowledge, this is the only existing method capable of translated alignment while retaining single-cell resolution. The small size of PalmDB (36 MB) enables efficient detection of orders of magnitude more viruses than detection based on (NCBI) reference genomes. Moreover, operating in the amino acid space yields a method robust to silent nucleotide mutations.



```
# 1. Install kb-python (optional: install gget to fetch the host genome and/or transcriptome)
pip install kb-python gget

# 2. Create reference index
# Optional: masking of host genomic and/or transcriptomic sequences using the D-list
# Single-thread runtime: 1.5 h; Max RAM: 4.4 GB; Size of generated index: 593 MB
# Without D-list: Single-thread runtime: 3.5 min; Max RAM: 3.9 GB; Size of generated index: 592 MB
kb ref \
    --aa \
    --d-list $(gget ref --ftp -w dna homo_sapiens) \
    -i index.idx --workflow custom \
    palmdb_rdrp_seqs.fa

# 3. Align sequencing reads and generate count matrix
# Single-thread runtime: 1.5 min per 1 million sequences; Max RAM: 2.1 GB
kb count \
    --aa \
    -i index.idx -g palmdb_clustered_t2g.txt \
    --parity single \
    -x default \
    user_data.fastq.gz
```

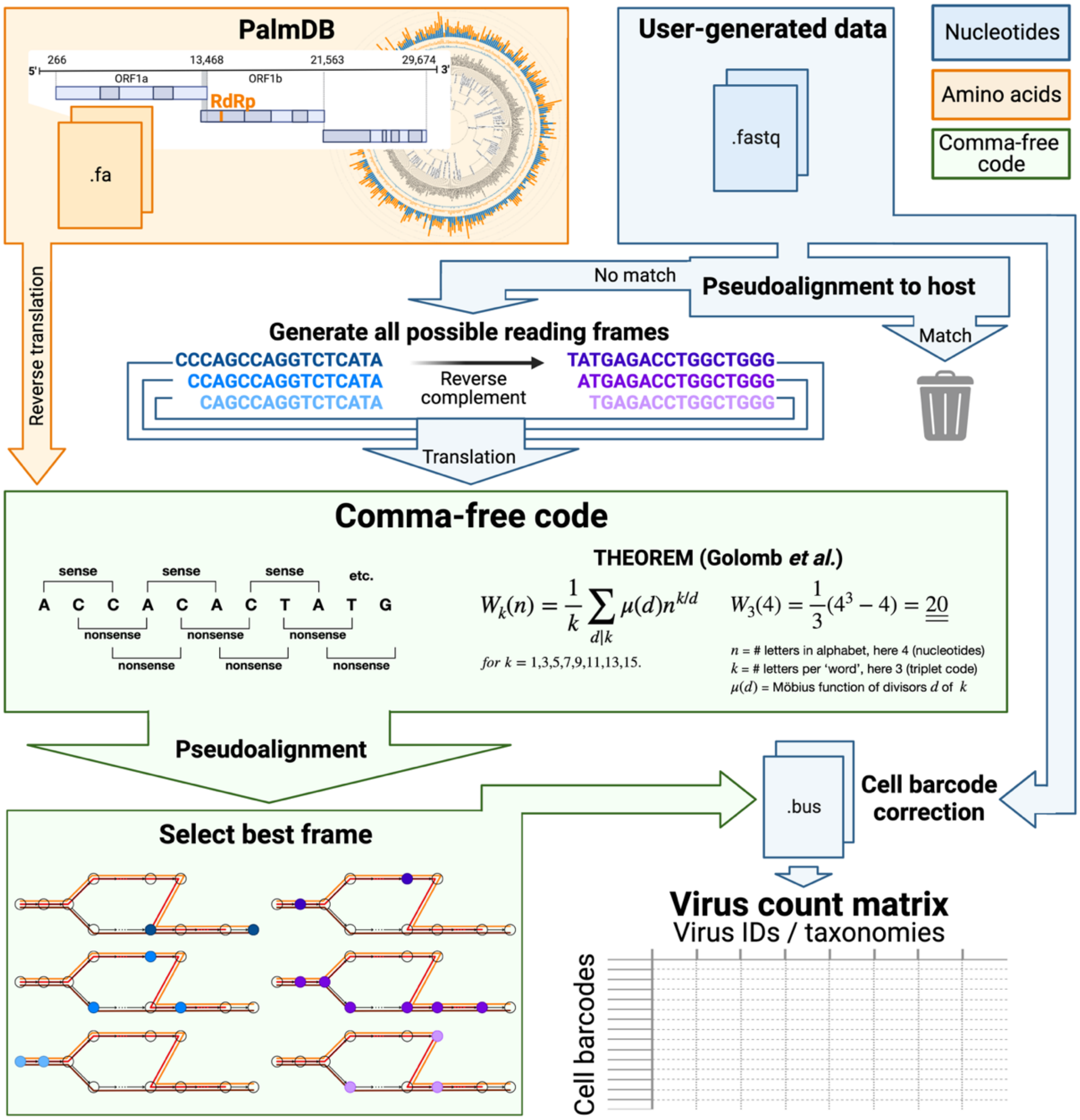


Figure legend on next page.

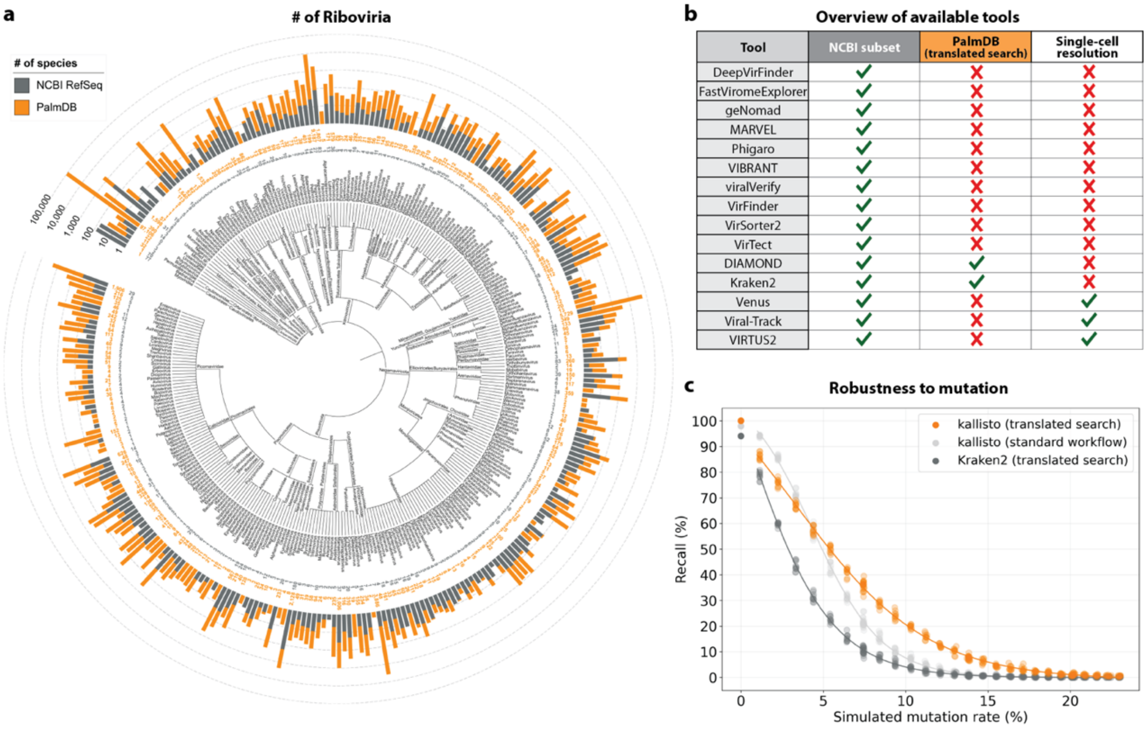
Fig. 3.1: Schematic overview of the kallisto translated search front- and back-end. The front-end is similar to kallisto-bustools workflows as previously described[43,44]. The user provides sequencing data, usually in the form of FASTQ files, as well as a reference FASTA file containing amino acid sequences to align the nucleotide sequencing data against. The novel argument '--aa' activates the translated search alignment. In the example shown here, the reference file consists of the PalmDB amino acid RdRP sequences contained in 'palmdb_rdrp_seqs.fa.' The 'palmdb_clustered_t2g.txt' file groups virus IDs with the same taxonomy across all main taxonomic ranks like transcripts of the same gene (see Methods). Both files are available here: https://tinyurl.com/4wd33rey. During the generation of the reference index with 'kb ref', the D-list option may be used to mask host genomic and/or transcriptomic sequences, as further discussed in this manuscript. Here, human genomic sequences fetched from Ensembl using gget[45] are masked using the D-list. The reference index only needs to be generated once, and precomputed PalmDB reference indices for human and mouse hosts are available here: https://tinyurl.com/aaxyy8v8. Following the generation of a reference index, the sequencing reads are pseudoaligned to the reference index, and a count matrix is generated using the 'kb count' command. The '-x' argument is used to define the sequencing technology. In the example code, the minimum required user input is marked in orange (amino acid space) and blue (nucleotide space). In the kallisto translated search back-end, the reference amino acid sequences and the nucleotide sequencing reads are translated into a non-redundant comma-free code. For the nucleotide sequences, the translation occurs in all six possible reading frames (three forward and three reverse frames). The pseudoalignment is performed in the comma-free code space and is compatible with the kallisto cell barcode tracking which enables analysis at single-cell resolution. The workflow generates a cell barcode by virus ID count matrix.

Results

Translated alignment of nucleotide sequences to an amino acid reference with kallisto enables efficient, accurate detection of RNA viruses in transcriptomic data at single-cell resolution

Most existing methods to detect viral sequences either (i) rely on and are limited to (NCBI) reference genomes[14–23], (ii) are not able to perform translated alignment of nucleotide data to an amino acid reference[24–26], or (iii) are unable to retain single-cell resolution through cell barcode tracking[27,28] (Fig. 3.2b). We expanded the bulk and single-cell RNA-seq preprocessing tool kallisto[13] to allow translated search and validated its use in combination with PalmDB for the detection of virus-like sequences in single-cell and bulk RNA sequencing data. PalmDB is a database of 296,623 unique RdRP-containing amino acid sequences, representing 146,973 virus species[9]. Fig. 3.2a provides an overview of the number of entries per taxonomy in the NCBI and PalmDB databases. The figure can also be viewed interactively here: tinyurl.com/4dzwz5ny.

The translated alignment is performed by first reverse translating the amino acid reference sequences and all possible reading frames (three forward and three reverse) of the nucleotide sequencing reads to comma-free code (Fig. 3.1)[29]. A comma-free code is a set of k-letter 'words' selected such that any off-frame k-mers formed by adjacent letters do not constitute a 'word', and will thus be interpreted as 'nonsense'. For k=3 (a triplet code) and 4 letters (e.g. 'A', 'T', 'C', and 'G'), this results in exactly 20 possible words (theorem shown in Fig. 3.1), which equals the number of amino acids specified by the universal genetic code. Due to the serendipity of these numbers, Crick *et al.* hypothesized the genetic



Tool	NCBI subset	PalmDB (translated search)	Single-cell resolution
DeepVirFinder	✓	✗	✗
FastViromeExplorer	✓	✗	✗
geNomad	✓	✗	✗
MARVEL	✓	✗	✗
Phigaro	✓	✗	✗
VIBRANT	✓	✗	✗
viralVerify	✓	✗	✗
VirFinder	✓	✗	✗
VirSorter2	✓	✗	✗
VirTect	✓	✗	✗
DIAMOND	✓	✓	✗
Kraken2	✓	✓	✗
Venus	✓	✗	✓
Viral-Track	✓	✗	✓
VIRTUS2	✓	✗	✓

Fig. 3.2: a, Phylogenetic tree of the taxonomies of viral sequences/genomes included in the PalmDB sOTUs and NCBI RefSeq databases from phylum to genus. Barplots indicate the number of sequences/species available for each taxonomy in each database. The tree was generated with iTOL[46]. This plot can also be viewed interactively here: tinyurl.com/4dzwz5ny. **b**, Overview of available tools for the detection of viral sequences in next-generation sequencing data[14–26,28,47], and their ability to align to NCBI RefSeq nucleotide genomes, perform translated alignment of nucleotide data against an amino acid reference, and retain single-cell resolution through cell barcode tracking. **c**, Mutation-Simulator[48] was used to add random single nucleotide base substitutions to 676 ZEBOV RdRP sequences obtained by Seq-Well sequencing[37] at increasing mutation rates. We performed 10 simulations per mutation rate. The sequences were subsequently aligned using kallisto translated search against the complete PalmDB, Kraken2 translated search against the RdRP amino acid sequence of ZEBOV with a manually adjusted NCBI Taxonomy ID to allow compatibility with Kraken2, and kallisto standard workflow against the complete ZEBOV nucleotide genome (GCA_000848505.1). The plot shows the recall percentage of the 676 sequences for each of the 10 simulations at each mutation rate. Each was fitted with an inverse sigmoid for mutation rates > 0.

code to be a comma-free code in 1957[30]. The impossibility of off-frame matches makes comma-free codes highly appropriate for translated alignment (Extended Data Fig. 3.10, Methods). The *de Bruijn* graph generated from the reverse translated PalmDB sequences groups viruses of the same taxonomies, indicating that within-taxonomy similarity is conserved in comma-free space as expected (Extended Data Fig. 3.4d). Finally, the six reading frames of the sequencing reads translated to comma-free code are pseudoaligned to the reference sequences reverse translated to comma-free code. If several reading frames of the same read produce alignments, the best frame is chosen (Fig. 3.1, Methods).

The workflow can be executed in three lines of code, and computational requirements do not exceed those of a standard laptop (Fig. 3.1). Building on kallisto's versatility, the

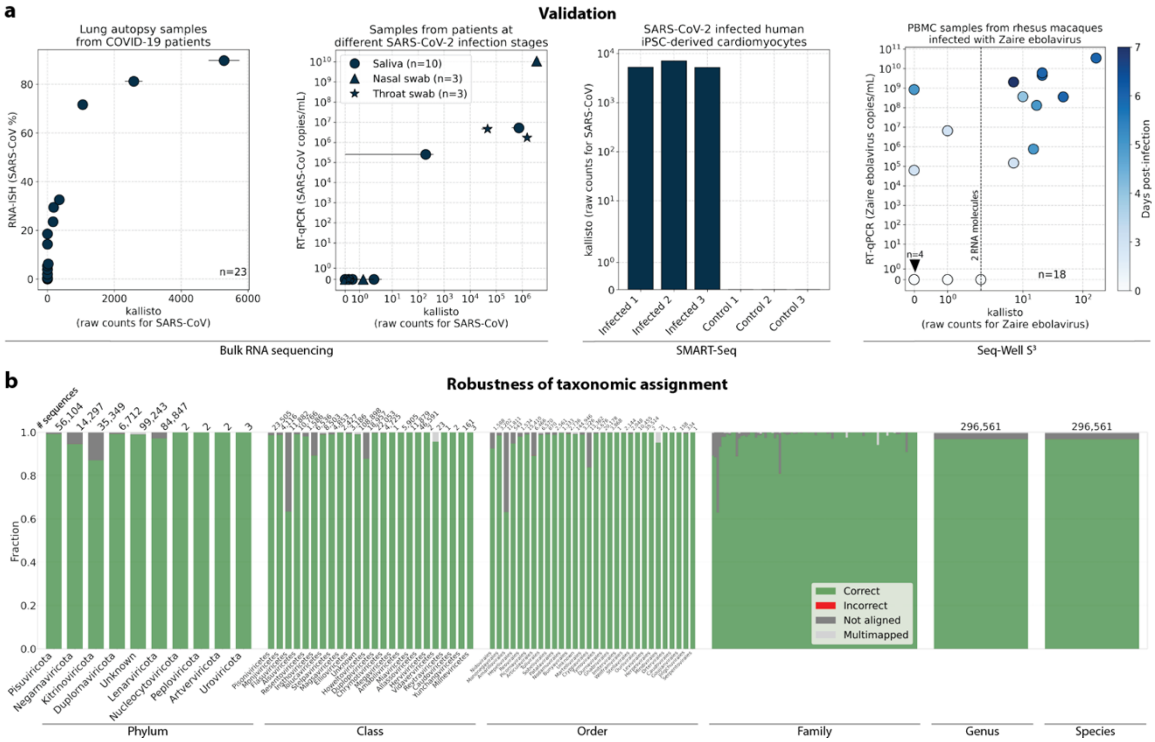


Fig. 3.3: a, Sequencing data from samples with a known viral infection and sequenced using different bulk and single-cell RNA sequencing technologies was aligned to PalmDB using kallisto translated search. Viral load obtained through alternative methods, such as RNA-ISH and qPCR, is compared to the target virus counts returned by kallisto. From left to right: 1. RNA-ISH (%) over total raw kallisto counts for SARS-CoV for 23 lung autopsy samples from COVID-19 patients obtained by bulk RNA sequencing[34]. Error bars show min-max values for each read in a pair; the dot shows the mean. 2. SARS-CoV-2 viral load by RT-qPCR (copies/mL) over total raw kallisto counts for SARS-CoV species obtained by bulk RNA sequencing of 16 saliva (circle), nasal swab (triangle), and throat swab (star) specimens from patients with acute SARS-CoV-2 infection[35,36]. Each specimen underwent duplicate library preparation and paired-end sequencing; points indicate the mean among the paired reads and duplicates, and error bars show min-max values. 3. Total raw kallisto counts for SARS-CoV species for 3 human iPSC-derived cardiomyocytes infected with SARS-CoV-2 and 3 control samples obtained by SMART-Seq[38]. 4. RT-qPCR (copies/mL) over total raw kallisto counts for ZEBOV for 19 rhesus macaque blood samples obtained during different stages of infection with ZEBOV and sequenced with Seq-Well[37]. **b**, To validate the mapping of nucleotide sequences to an amino acid reference with kallisto translated search and assess the robustness of the taxonomic assignment, we reverse translated all amino acid sequences in the PalmDB using the 'standard' genetic code (see Methods). The reverse translated PalmDB RdRP sequences were subsequently aligned to the optimized PalmDB amino acid reference (see Methods) with kallisto translated search. For each sequence, we differentiated the mapping result at each taxonomic rank into four categories: 'correct' or 'incorrect' taxonomic assignment based on the sOTU to virus ID mapping, 'multimapped' (the sequence aligned to multiple targets in the reference and could not unambiguously be assigned to one), or 'not aligned' (the sequence was not aligned). The plot shows the fraction of sequences falling into each mapping result category assessed at each taxonomic rank. The numbers above the bars indicate the total number of sequences per rank. Family names and numbers were omitted, and genera and species ranks were combined for readability.

workflow is compatible with all state-of-the-art single-cell and bulk RNA sequencing methods, including but not limited to 10x Genomics, Drop-Seq[31], SMART-Seq[32], SPLiT-

Seq[33] (including Parse Biosciences), and spatial methods such as Visium.

Validation testing was performed using different bulk and single-cell RNA sequencing datasets with known infections with severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) or *Zaire ebolavirus* (ZEBOV)[34–38]. In these tests, translated search with kallisto and PalmDB was able to detect the viral RNA and correctly assign species-level taxonomy at counts correlating with viral loads measured by RT-qPCR or RNA-ISH, regardless of the technology used to generate the data (Fig. 3.3a). Fig. 3.3b provides an overview of the robustness of the taxonomic assignment across all available taxonomic ranks after reverse translated RdRP sequences were aligned to the PalmDB with kallisto translated search. At the species level, 96.76 % of 296,561 sequences were assigned the correct taxonomy, 0.007 % were assigned an incorrect taxonomy, 0.37 % could not be unambiguously matched to a single virus (they were multimapped), and 2.86 % were not aligned. This confirms that the sequence transformation introduced by the kallisto translated search pipeline retains taxonomic assignments with up to species-level specificity.

Next, we sought to confirm that kallisto translated search with PalmDB correctly identifies sequences that originate from the RdRP gene. To this end, we selected a subset of 100,000,000 reads obtained using Seq-Well sequencing of macaque peripheral blood mononuclear cell (PBMC) samples obtained at 8 days post-infection with ZEBOV[37] (see Methods). We aligned the reads to the PalmDB amino acid sequences with kallisto translated search. We also aligned the reads to the complete ZEBOV nucleotide genome using Kraken2 (standard nucleotide alignment)[27]. Aligned reads from both alignments were extracted and realigned to the ZEBOV genome using bowtie2[39], a BAM file was created using SAMtools[40] and the alignment was subsequently visualized NCBI Genome Workbench[41]. The visualized alignments are shown in Extended Data Fig. 3.2 and confirmed that kallisto translated search accurately and comprehensively detected ZEBOV RdRP sequences.

We then tested whether our translated search method is robust to single nucleotide mutations, which occur at a relatively high rate in RNA viruses of up to 10^{-4} substitutions per nucleotide site per cell infection[42]. We added random single nucleotide base substitutions to 676 ZEBOV nucleotide RdRP sequences identified during the alignment described in the previous paragraph[37], then assessed the frequency of correct taxonomic classification (recall percentage) by kallisto translated search, in comparison to the current state-of-the-art translated search tool, Kraken2 (translated search). kallisto translated search correctly recalled up to 27.5 % more viral RdRP sequences than Kraken2 (translated search) (Fig. 3.2c). Moreover, kallisto translated search was more robust than aligning to the complete nucleotide genome with the standard kallisto workflow at mutation rates > 4 % (Fig. 3.2c), which emphasizes the advantage of operating in the amino acid space. While the Kraken2 (translated search) and the kallisto standard workflow were given only the correct virus as a reference (here, ZEBOV), kallisto translated search had to distinguish between all viruses contained in the PalmDB and identify the correct taxonomy. kallisto translated search was able to maintain > 90 % precision in the species-level taxonomic assignment at mutation rates up to 12 % (Extended Data Fig. 3.4b).

We next sought to investigate whether viral species not included as species-like operational taxonomic units (sOTUs) in the reference PalmDB database could be detected based on the conservation of the RdRP gene. To do this, we removed all *Ebolavirus* species, all *Ebolavirus* genera, and all members of the *Filoviridae* family from the reference, and subsequently aligned the 676 ZEBOV RdRP sequences obtained by Seq-Well sequencing37. In each scenario, a subset of sequences aligned to the nearest remaining relative based on the main taxonomic rank (Extended Data Fig. 3.1). This suggests that kallisto translated search can detect the highly conserved RdRP of a large number of viral species, beyond the number of sequences in the PalmDB database, while still providing reliable sOTU-based taxonomic assignment of lower-rank taxonomies.

Read and virus filtering

A common problem that arises during the identification of microbial sequences is the cross-species contamination of reference genome databases, such as the ubiquitous contamination of bacterial genomes with human DNA[49–51]. The PalmDB is not a curated database, and it is possible that some virus-like sequences in the PalmDB are not derived from viruses. This can lead to the misclassification of host reads as bacterial or viral, suggesting the presence of microbes that were not truly present. The misclassification of host reads as viral can be prevented by removing host reads prior to the alignment to the viral reference. However, conservatively removing host reads will also remove sequences of endogenous viral elements, which are very abundant in vertebrate genomes[52] and may lead to the removal of viral sequences that were truly present. Hence, there are two goals: (i) removing host reads to prevent the misclassification of host reads as viral while (ii) comprehensively identifying the virome within a sample.

We first evaluated the impact of different host masking options on the resulting virome. We used kallisto translated search with PalmDB to map the virus profiles of peripheral blood mononuclear cell (PBMC) RNA sequencing samples from 19 rhesus macaques and applied different host masking workflows. The approach to masking host versus microbe reads and the handling of overlap between reference sequences can affect the downstream result. For example, sequences with varying sizes of virus-host overlap, sequences that span the junction of two exons, and entirely ambiguous sequences can influence the outcome of the masking and generate highly variable results depending on the method used (Extended Data Fig. 3.4a and 5). Depending on the research question and design, any one or a combination of different masking options might be appropriate. We explored the following masking options, listed from least to most conservative:

No mask

We aligned the sequencing reads to the PalmDB with kallisto translated search without masking or previously removing host sequences. For the macaque PBMC dataset, this masking option resulted in 243 distinct sOTUs detected (Fig. 3.4a).

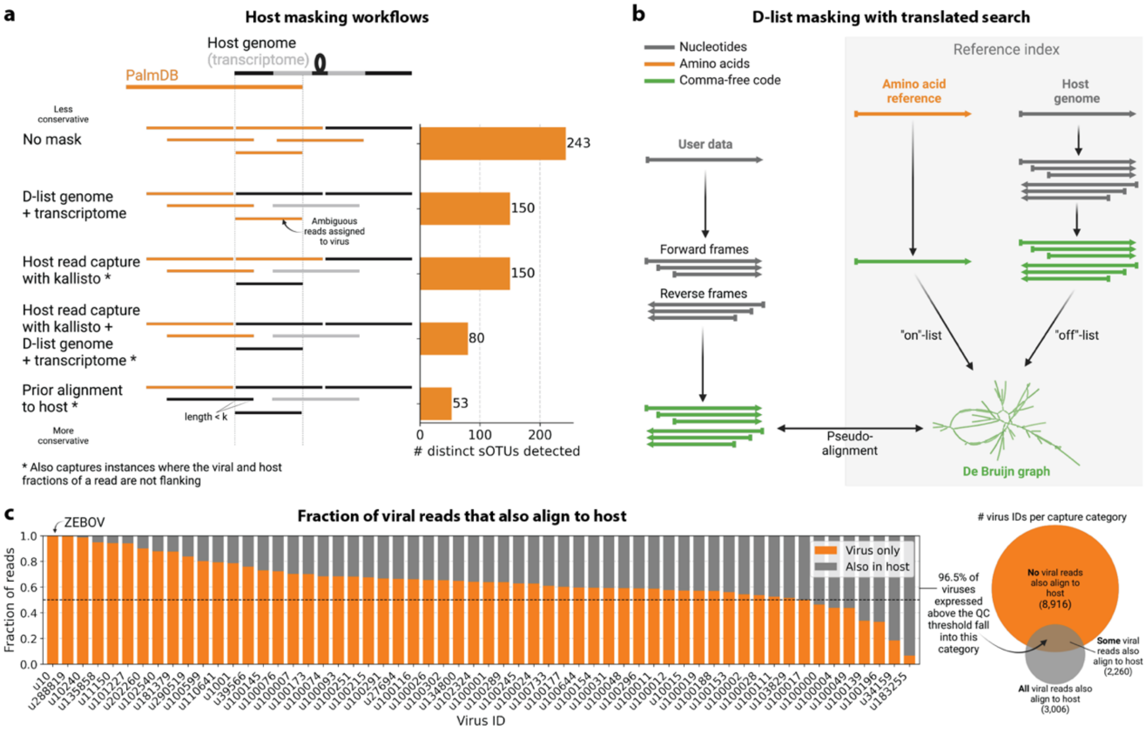


Fig. 3.4: a, Schematic overview of the different host masking options discussed in this manuscript. Reads that align to PalmDB and are considered viral are marked in orange and reads that align to the host genome or transcriptome are marked in black or grey, respectively. The barplot shows the number of distinct sOTUs, defined by distinct virus IDs observed in ≥ 0.05 % of cells for each workflow. **b**, Schematic overview of masking the host genome with the D-list argument when used in combination with translated search. The D-list genome consists of nucleotide sequences and hence is translated to comma-free code in all six possible reading frames, similar to the translation of the nucleotide sequencing reads. **c**, Masking host sequences with the kallisto read capture workflow generates two distinct virus count matrices: The first contains viral reads that only aligned to the PalmDB, and the second contains viral reads that aligned to the host transcriptome in addition to the PalmDB. The majority of viruses detected above the quality control (QC) threshold (observed in ≥ 0.05 % of cells), had reads that aligned to the host transcriptome as well as the PalmDB. The barplot shows the fraction of reads for each virus that aligned to the PalmDB only ('virus only') and those that aligned to the host transcriptome in addition to the PalmDB ('also in host').

D-list genome + transcriptome

To incorporate host read masking into our kallisto workflow, we quantified the reads while masking the host genome and transcriptome using an index created with the D-list (distinguishing list) option[53]. This option identifies sequences that are shared between a target transcriptome and a secondary genome and/or transcriptome. *k*-mers flanking the shared sequence on either end in the secondary genome are added to the index *de Bruijn* graph. During pseudoalignment, the flanking *k*-mers are used to identify reads that originated from the secondary genome but would otherwise be erroneously attributed to the target transcriptome due to the spurious alignment to the shared sequences. In our experiments, the target transcriptome consisted of the viral RdRP amino acid sequences contained in the PalmDB, and the secondary genome consisted of transcriptomic and genomic macaque and dog nucleotide sequences. When combining D-list with translated search, the secondary genome is translated to comma-free code in all six possible reading frames (Fig. 3.4b). This masking option can be easily added to the kallisto translated search workflow without any additional commands (Fig. 3.1). Masking the host transcriptome and genome with D-list resulted in 150 distinct sOTUs detected (Fig. 3.4a). Note that masking both the transcriptome and the genome, or either one will generate different results because masking only the genome will not mask sequences that span exon-exon junctions (Extended Data Fig. 3.4a and 3.5).

Host read capture with kallisto
To imitate prior alignment to the host genome, as performed with bwa (described below), within a simple, efficient kallisto workflow, we captured all reads that pseudoaligned to the host transcriptome with kallisto. Masking by capturing these host reads resulted in the same number of distinct sOTUs detected as masking with D-list (Fig. 3.4a).

Host read capture with kallisto + D-list genome + transcriptome
Although masking with D-list and capturing reads that aligned to the host transcriptome resulted in the same number of distinct sOTUs detected, the two methods masked different reads and resulted in different virus profiles (Fig. 3.5a, Extended Data Fig. 3.5). We decided to combine the D-list and host read capture masking approaches to achieve a conservative result similar to that achieved by prior alignment with bwa. In this approach, the sequencing reads were aligned to the PalmDB index with a D-list containing the host genome and transcriptome, and subsequently reads that pseudoaligned to the host transcriptome were captured. Combining the D-list and host read capture masking options reduced the number of detected sOTUs to 80 (Fig. 3.4a).

Prior alignment to host with bwa
We aligned the sequencing reads to the macaque and dog genomes using the highly sensitive alignment algorithm bwa[54] and removed all reads that aligned anywhere in the host genomes before alignment to PalmDB with kallisto translated search. This achieved very conservative masking of the host genome. However, this workflow is complex, time-consuming, and computationally expensive (~4.5 days using 60 cores for the macaque ZEBOV PBMC dataset). This workflow resulted in the detection of 53 distinct sOTUs (Fig. 3.4a).

There are inherent differences between these masking methods which are illustrated in Fig 4a and Extended Data Fig. 3.4a. Although the genome is passed to the software, the standard kallisto workflow builds an index based on the host transcriptome, not the entire host genome, since for genomes as large as the macaque genome, building the index on the entire genome would require a large amount of memory. Hence, masking by capturing reads that pseudoaligned to the host with kallisto will only capture host reads from mature mRNA molecules. If the D-list is passed both the transcriptome and the genome, it will be able to mask mature and nascent RNA molecules as well as RNA molecules originating from intergenic regions. The D-list index avoids excessive memory requirements by restraining the index to distinguishing sequences between viral and host sequences. As a result, reads that contain non-flanking host and viral sequences will not be filtered. Moreover, the D-list will favor viral assignment in the case of an entirely ambiguous read.

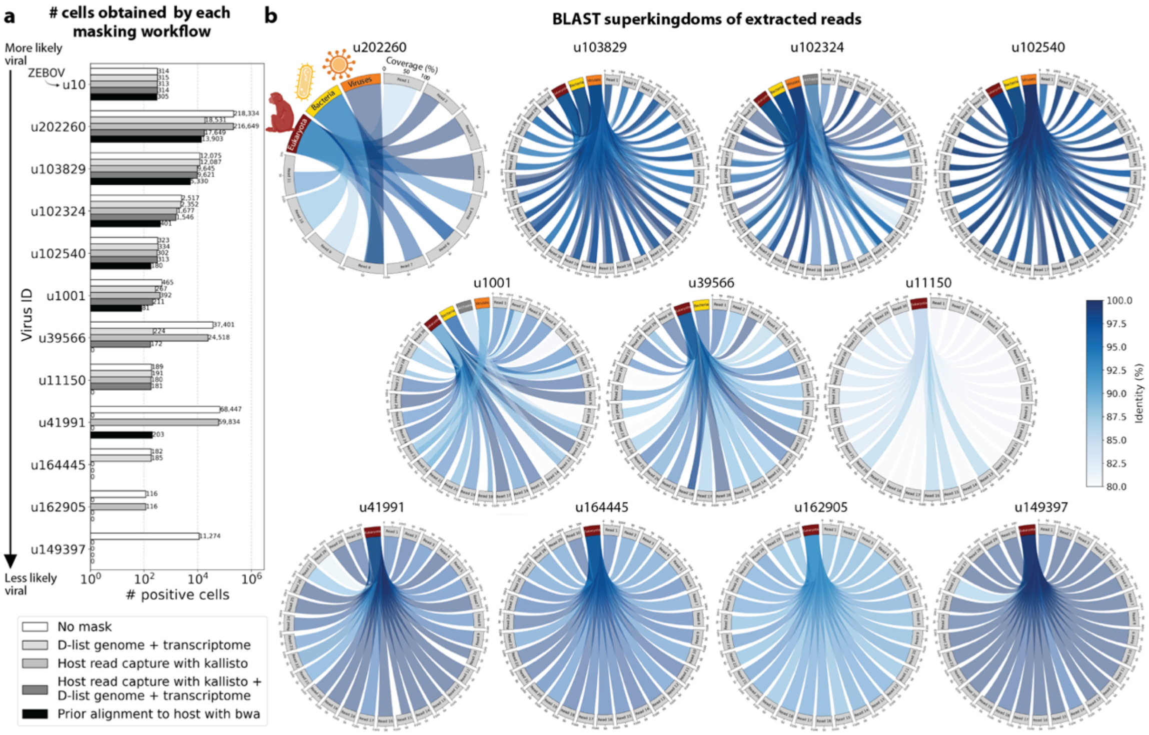


Fig. 3.5: a, The number of positive cells obtained for 12 different virus IDs by each masking workflow. Each masking workflow is described in detail in the Methods section. The cell counts for all viruses detected above the QC threshold for all masking workflows are shown in Extended Data Fig. 3.5. **b**, pyCirclize plots showing the BLAST+[55,56] results of randomly selected sequencing reads for each of the novel viruses shown in a (excluding the known virus ZEBOV which corresponds to virus ID 'u10'). Each circular plot corresponds to the results for one virus ID. Each light grey sector corresponds to one sequencing read that links to the superkingdoms (red (eukaryotes), yellow (bacteria), orange (viruses), and dark grey (archaea) sectors) based on its BLAST+ alignment results. The width of the connecting link indicates the BLAST+ alignment coverage percentage, and its color indicates the identity percentage. For u202260, approximately ⅔ of the extracted reads yielded no BLAST results.

Neither of these issues applies to masking with bwa since the alignment with bwa was performed against the host genome. Since bwa uses a smaller seed length than kallisto's default *k*-mer size of 31, bwa provides more sensitive alignment of sequencing reads against the host genome and provides the most stringent filtering.

To confirm that reads identified as viral were not misaligned macaque reads, we extracted randomly selected sequencing reads from 11 virus IDs and aligned them against the nucleotide sequence database with BLAST+[55,56] (Fig. 3.5b). The reads associated with virus IDs identified by all masking workflows (u202260, u103829, u102324, u102540, and u1001) BLAST-aligned with relatively low coverage and identity to several superkingdoms, including viruses. For u202260, approximately ⅔ of the extracted reads yielded no BLAST results. Given that the majority of RdRP sequences in the PalmDB originate from unknown viruses lacking reference genomes, it was expected that these sequences would not yield confident BLAST results. However, given the comprehensiveness of the macaque genome[57], misaligned macaque sequences should

BLAST to the macaque genome with high coverage and identity. The next two virus IDs, u39566 and u11150, were filtered out by the bwa workflow and did not BLAST to the viruses superkingdom. However, their BLAST results displayed relatively low coverage and identity, which would not be expected from macaque sequences. Below, we provide further evidence that u11150 sequences might have originated from an ongoing viral infection. This was likely an instance where filtering with bwa was too conservative and threw out viral sequences. u41991 was identified as viral by the bwa workflow but filtered out by the D-list + host capture workflow. Based on the BLAST results for u41991, which include high coverage and identity matches for eukaryotes, it is likely that filtering is the appropriate action. u164445 and u162905 were filtered by either capturing the host reads or using the D-list, respectively, and BLAST to eukaryotes with high coverage and identity, illustrating that a combination of the two methods leads to more robust results. Finally, sequences identified as u149397, which were filtered by all masking options and are only retained without masking, BLAST to eukaryotes with high coverage and identity.

Separately from exploring the results of different read masking options, we also investigated the question of virus filtering. Host read capture with kallisto generates two separate count matrices: One contains counts for reads that are solely viral, and a second contains counts for viral reads that also pseudoaligned to the host transcriptome. The distinction between filtering reads and filtering viruses becomes evident when examining the two count matrices: for the macaque PBMC dataset, we found that most viruses found in ≥ 0.05 % of cells had at least some reads that also mapped to the host transcriptome, including reads for ZEBOV (Fig. 3.4c and 3.5a). Moreover, aligning without host masking often led to the detection of more positive cells (Extended Data Fig. 3.5). Hence, naive masking of reads can lower the detection sensitivity of viruses that seem truly present. Our masking workflows facilitate the identification of viruses with a high likelihood of being truly present based on conservative host read masking, while also obtaining unmasked reads for these viruses to prevent the decrease in sensitivity inherent to masking host reads. We applied this approach when training the logistic regression models described below to minimize the occurrence of false viral absence.

The presence of novel putative viruses perturbs host gene expression in macaque blood cells, allowing prediction of viral presence based on host gene expression at single-cell resolution

We used kallisto translated search and the PalmDB to map the viral profiles of PBMC samples from 19 rhesus macaques sequenced at different stages of Ebola virus disease (EVD)[37] (Fig. 3.6a) at single-cell resolution. The dataset consisted of 30,594,130,037 reads in total. After alignment to both the host genome (using the standard kallisto workflow) and PalmDB (using kallisto translated search with D-list + host capture masking), and quality control using the host count matrix (Extended Data Fig. 3.3a, Methods), we retained 202,525 PBMCs. We used the Leiden algorithm[58] to partition the PBMC transcriptomes into 18 clusters of similar macaque gene expression, of which 16 could be assigned cell types based on common marker genes (Extended Data Fig. 3.3d).

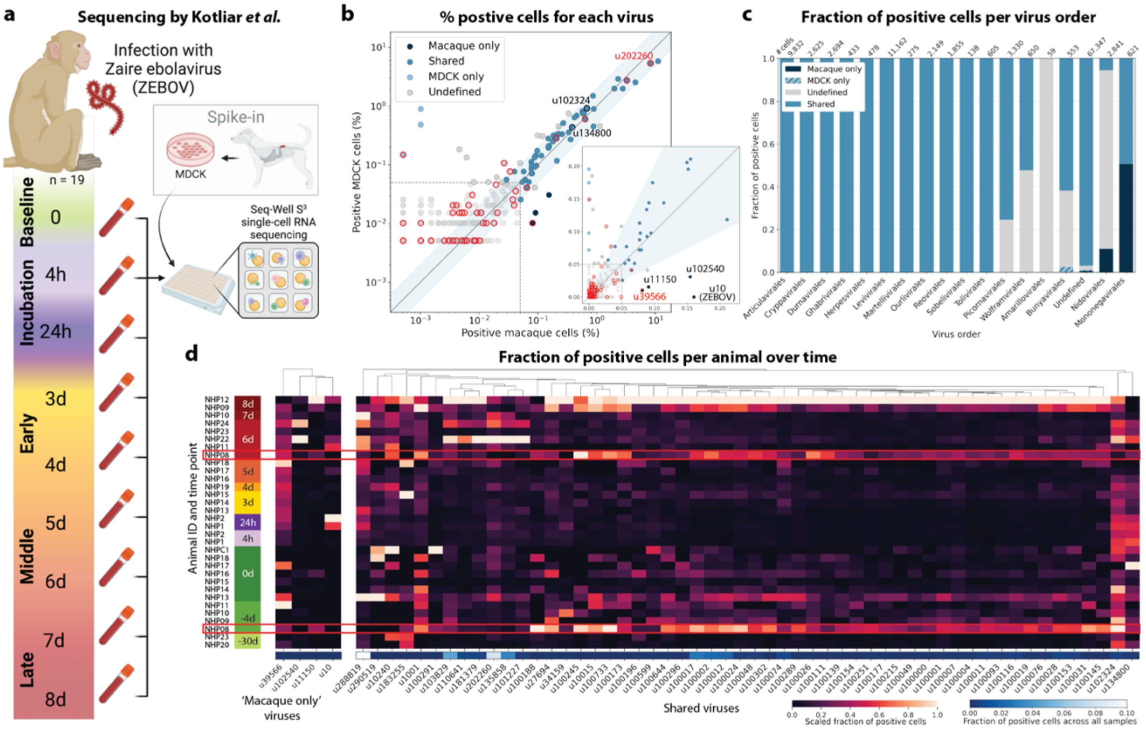


Fig. 3.6: **a**, Schematic overview of the single-cell RNA sequencing data collected by Kotliar *et al.*[37]. Kotliar *et al.* performed single-cell RNA sequencing of peripheral blood mononuclear cell (PBMC) samples from 19 rhesus macaques at different time points during Ebola virus disease (EVD) after infection with *Zaire Ebolavirus* (ZEBOV) using Seq-Well[74] with the S3 protocol[75]. A subset of the PBMC samples were spiked with Madin-Darby canine kidney (MDCK) cells. This schematic was adapted from the original design by Kotliar *et al.* **b**, For each virus-like sequence, the percentage of positive MDCK cells is plotted against the percentage of positive macaque cells. Virus IDs were categorized into 'shared', 'macaque only', 'MDCK only', and 'undefined' as described in the Methods section. The insert shows the same plot without log scale axes such that zero counts are included. A red edge marks contaminating virus-like sequences also observed in sequencing data obtained from blank sequencing libraries containing only sterile water and reagent mix (Extended Data Fig. 3.9c). **c**, Bar plot showing the fraction of positive cells obtained for each viral order, as defined by the PalmDB sOTUs, for each category. **d**, Fraction of positive cells for all 'macaque only' and 'shared' virus IDs. Each row corresponds to one animal at a specific EVD time point. The fractions were scaled to range from zero to one for each virus ID. The raw total fraction of positive cells for each virus ID across all samples is shown in blue below.

The obtained cell types, their marker gene expression and relative abundance over time are consistent with the results reported by Kotliar *et al.*[37], including the emergence of a cluster of immature neutrophils and decreased lymphocyte abundance, especially natural killer cells, during EVD (Fig. 3.7a). While density based PBMC isolation typically removes neutrophils, immature neutrophils are less dense than mature neutrophils and can co-isolate with PBMCs during infections[37]. Clusters of the same cell type were often separated by time point (Fig. 3.7a), indicating changes in macaque gene expression within the same cell type over the course of the EVD. This is in agreement with results obtained by mass cytometry in Kotliar *et al.*[37].

ZEBOV count data from this analysis workflow was also consistent with previously reported results. Since only a small fraction of the RNA molecules in these tissue samples are viral and of those, we only detect the RdRP, the measured absolute RNA counts for any one virus per cell are low (Extended Data Fig. 3.3e). As a result, we converted the virus count matrix into a binary matrix where each virus was recorded as being either present or not present in each cell. This approach has been previously validated for sparse single-cell RNA, specifically viral, sequencing data[24,59], and prevented the need for further normalization by individual cellular viral load, which may introduce biases[49]. The presence of virus in each cell was then used to determine the viral abundance among populations of cells composed of clusters, cell types, or tissues. First, we used the binary virus count matrix to validate the detection of ZEBOV. Samples obtained during incubation displayed the highest abundance of ZEBOV-positive cells, and ZEBOV-positive cells remained detectable at all following time points (Fig. 3.7b, top left). These trends are consistent with the results reported by Kotliar *et al.*[37].

The parallel analysis of viral and host gene counts at single-cell resolution allowed the identification of infected cell types based on host gene expression to reveal that ZEBOV-positive cells consisted predominantly of monocytes (Fig. 3.7b, top right). These results are consistent with previous literature on ZEBOV tropism[60] and reproduce the ZEBOV abundance trends obtained by alignment to the ZEBOV genome[37]. This indicates that while the total viral counts obtained by kallisto translated search with PalmDB are low due to only detecting the RdRP, comparative trends are captured accurately. All *Ebolavirus* reads were identified correctly as ZEBOV with no counts detected for other *Ebolavirus* species (Extended Data Fig. 3.6).

Our analysis workflow identified virus-like sequences with sOTUs other than ZEBOV in this dataset. These virus-like sequences may be present due to, amongst others, viral infection of the host, host endogenous viral elements, infection of microbes residing in the host, infection of food ingested by the host, or laboratory contamination. Fig. 3.7b (bottom left and right) shows the total number of distinct sOTUs (corresponding to distinct virus IDs) detected over time and per cell type. We observed a slight increase in the number of distinct sOTUs detected per cell during the later stages of EVD, driven by T cell, B cell, and neutrophil clusters with high fractions of cells during later EVD stages (Fig. 3.7a). Neutrophils showed the highest numbers of distinct sOTUs per cell (Fig. 3.7b, bottom right). Since neutrophils fulfill their microbicidal function through phagocytosis and pinocytosis, it is possible that viral RNA was picked up by these cells through ingestion. In the following paragraphs, we explore different approaches to interpret the presence of these virus-like sequences.

Among the samples in this dataset, we detected a total of 11,176 virus-like sequences with at least one read that aligned to the PalmDB and did not align to the host (Fig. 3.4c), including many sOTUs from genera known to infect rhesus macaques (Extended Data Fig. 3.6)[61]. However, the majority of these virus-like sequences were expressed in less than 0.05 % of cells, which we defined as a quality control (QC) threshold (Fig. 3.6b, Methods). All of the virus-like sequences with positive cell fractions above the QC threshold in macaque

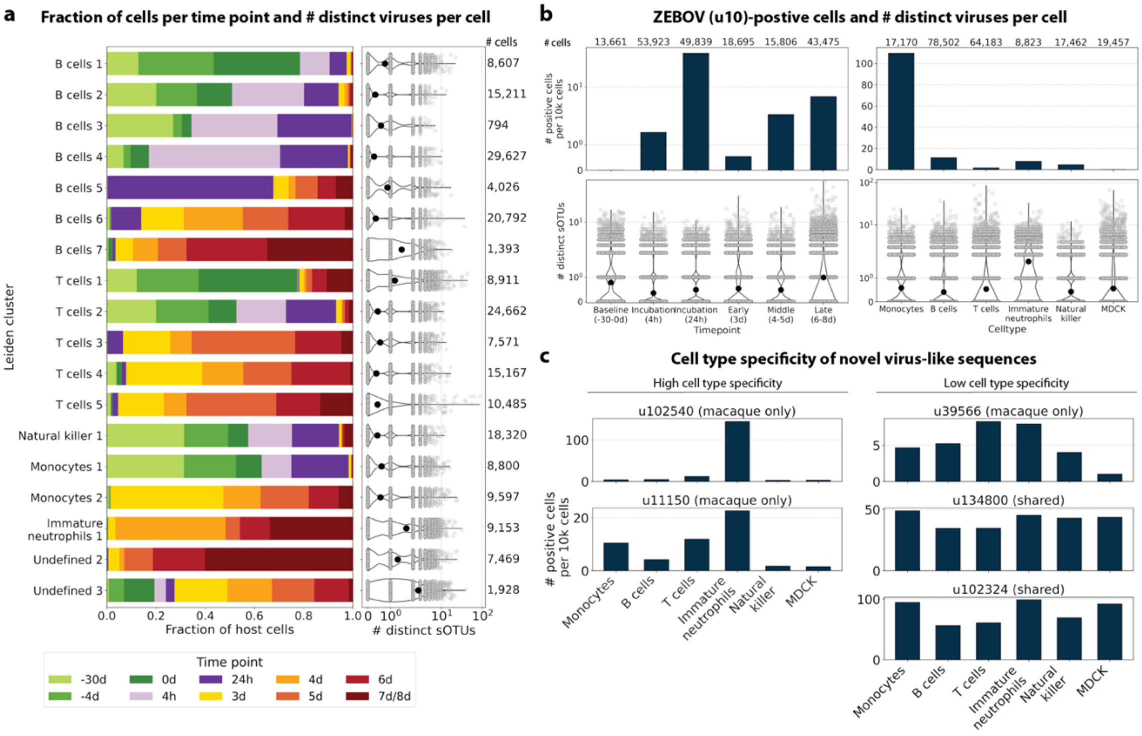


Fig. 3.7: a, The fraction of cells occupied by each EVD time point is shown per Leiden cluster. Each Leiden cluster was assigned a cell type based on previously defined marker genes (Extended Data Fig. 3.3d). On the right, the number of distinct sOTUs detected in each cell is shown. Each grey dot corresponds to one cell, and the black dot corresponds to the mean across all cells. **b**, The number of ZEBOV (u10) positive cells per 10,000 cells is plotted per EVD time point (left) and per cell type (right). For each time point and cell type, the number of distinct sOTUs found per cell is plotted at the bottom. Each grey dot corresponds to one cell, and the black dot corresponds to the mean across all cells. **c**, The number of positive cells per 10,000 cells is shown per cell type for the remaining three (excluding ZEBOV) macaque only virus IDs and two shared virus IDs. Virus IDs that show relatively high cell type specificity are shown on the left, and virus IDs with relatively even detection across all cell types are shown on the right.

cells that passed quality control can be explored in this interactive Krona plot[62] broken down by animal, timepoint, taxonomy, and fraction of cells occupied by each virus: https://tinyurl.com/23h6k36u.

A subset of samples included a spike-in of Madin-Darby canine kidney (MDCK) cells, resulting in a total of 23,500 MDCK cells after quality control and species separation (Extended Data Fig. 3.3b and c, Methods). We used the spike-in to further break down the viruses into ‘macaque only’, ‘MDCK only’, and ‘shared’ viruses (Fig. 3.6b, Methods). We expected that shared viruses occurring in both macaque and MDCK cells would include viruses introduced by the contamination of laboratory reagents used during sample preparation and sequencing[63], cell-free RNA contamination, endogenous retroviruses[52,64], and widespread latent infections. After filtering and categorization of viruses, we detected 4 (including ZEBOV) macaque only, 7 MDCK only, 15 undefined, and 54 shared viruses. This result suggests that the majority of virus-like sequences detected in this dataset were introduced through contamination. Indeed, many virus-like sequences that fell into the shared category could also be detected in 'blank' sequencing libraries containing only sterile water and reagent mix, providing evidence for their origin in widespread laboratory reagent contamination[63] (Fig. 3.6b, Extended Data Fig. 3.9c). The sOTUs of the macaque only and shared virus IDs, when available, are listed in Extended Data Table 1. Fig. 3.6c shows the fraction of reads occupied by each viral order (as defined in the sOTU for each virus ID) for macaque only, MDCK only, and shared viruses. Among viruses shared between macaque and MDCK cells, *Levivirales* (recently renamed to Norzivirales[65]), *Articulavirales* (which include the family of influenza viruses), and viruses of unknown taxonomy made up the largest fractions. *Norzivirales* are an order of bacteriophages, the majority of which were discovered in metagenomics studies[66]. They might have been introduced through bacterial contamination during sample preparation and sequencing. The shared viruses also included orders such as *Herpesvirales*, which are widespread, sometimes spreading through cross-species infections, and are known to persist in their host as latent infection[67,68]. Virus-like sequences detected in MDCK cells included sOTUs from the order of *Bunyavirales*, which infect a wide range of hosts, including MDCK cells[69], as well as virus-like sequences of unknown order. Virus-like sequences found only in macaque cells were of unknown order, in the order *Mononegavirales*, which includes ZEBOV, and in the order *Nidovirales*, which are known to infect mammals and include the family *Coronaviridae*. Virus-like sequences of known order (based on the sOTU) for each group are reasonably expected to be present in the respective sample types and the context of the hosts, which supports the biological validity of these viral read classifications.

To visualize the virus profiles of individual animals and over time, we plotted the fraction of positive cells for each macaque only and shared virus ID per animal and time point (Fig. 3.6d). The relative viral abundances varied, both between individual monkeys and time points. Notably, in some instances where the same animal was measured across several time points, the viral profile of this animal was reproduced in the later time point (Fig. 3.6d, Fig. 3.8a). The viral profiles of animal NHP08 at -4 days pre-infection and 6 days post-infection with ZEBOV are highlighted in the heatmap (Fig. 3.6d). Animals NHP1 and NHP2 each had two samples sequenced 20 hours apart which displayed highly similar viral profiles for each animal over time (Fig. 3.8a). This suggests that viral profiles sampled and sequenced within a short time window are coherent over time and across samples which is consistent with expectation. Several virus-like sequences, including u102324, were present in all animals and time points with relatively similar abundance (Fig. 3.6d, Extended Data Fig. 3.7a), coherent with their classification as shared sequences likely originating from contamination.

We then attempted to further determine which virus-like sequences were likely present due to viral infection of the host based on cell-type specificity and a coinciding host antiviral response. We visualized the viral tropism of the remaining three (other than ZEBOV) macaque only viruses. Two of them, u102540 and u11150, displayed relatively high sample-specificity while u39566 was abundant across all samples, similar to the shared viruses u134800 and u102324 (Extended Data Fig. 3.7a). The sOTU of u102540 indicates that it is an *Alphacoronavirus sp.*, which are known to infect rhesus macaques[61]. u102324 is predicted to belong to the family *Iflaviridae* (Extended Data Table 1), which is a family of viruses that infect insects[70], and the viral reads from this virus ID were likely not the result of an ongoing viral infection. The remaining virus IDs, u11150, u39566, and u134800, are of unknown taxonomy across all taxonomic ranks.

Two virus-like sequences exhibited cell-type specificity suggestive of viral infection. Of the three macaque only virus IDs excluding ZEBOV, we found that u102540 and u11150 showed high cell type specificity, while u39566 was expressed more evenly across all cell types (Fig. 3.7c). While u39566 was categorized as ‘macaque only’ above, it is likely a contaminating sequence given its presence in the blank sequencing libraries (Extended Data Fig. 3.9c). The lack of cell-type specificity coincides with u39566 sequences originating from reagent contamination and illustrates the importance of combining several different approaches, as described here, when interpreting the presence of virus-like sequences. u102540 (*Alphacoronavirus sp.*) exhibited high fractions of positive cells in neutrophils, while u11150 also displayed lower expression in monocytes, B cells, and T cells. Neutrophils play an important role in the innate immune response and promote virus clearance through phagocytosis. During phagocytosis, neutrophils engulf virions and apoptotic bodies. It is possible that the cell type specificity towards neutrophils observed here was due to neutrophils engulfing viral RNA during phagocytosis rather than viral tropism. As expected, the shared viruses u134800 and u102324 did not display cell type specificity (Fig. 3.7c).

The simultaneous analysis of the host and virus count matrices supported that several viruses identified were likely infecting the host and revealed virus-induced host gene expression. We hypothesized that viral presence in individual cells may be predicted based on the host gene expression. Since our workflow maintains single-cell resolution, we can analyze viral presence and host gene expression at single-cell resolution in parallel and investigate whether the presence of a virus affects host gene expression. We trained logistic regression models for all macaque only and shared (present in both MDCK and macaque cells) virus-like sequences to predict viral presence or absence in individual cells based on the cell’s host gene expression. The models were either trained on all or only highly variable macaque genes and with or without the donor animal and time point as covariates. After training models using a random selection of virus-positive and an equal number of virus-negative cells, we tested the model predictions on held-out test cells (Fig. 3.8b, Extended Data Fig. 3.8a and e). Given the cell type specificity of several of the virus-like sequences, virus-negative training cells were selected to be of the same cell types as virus-positive cells to ensure that we were not simply predicting cell type rather than viral presence.

We found that the presence or absence of virus-like sequences that displayed cell type- and sample-specificity (u10 (ZEBOV), u102540, and u11150) could be predicted at > 70 % accuracy (Fig. 3.8b and c), although for u11150, the sensitivity decreased with the inclusion of the covariates donor animal and EVD time point (Extended Data Fig. 8a). The sensitivity and specificity are shown in Extended Data Fig. 3.8a. By contrast, the presence of viruses that did not display cell type-specificity (u39566, u134800, and u102324) could not be

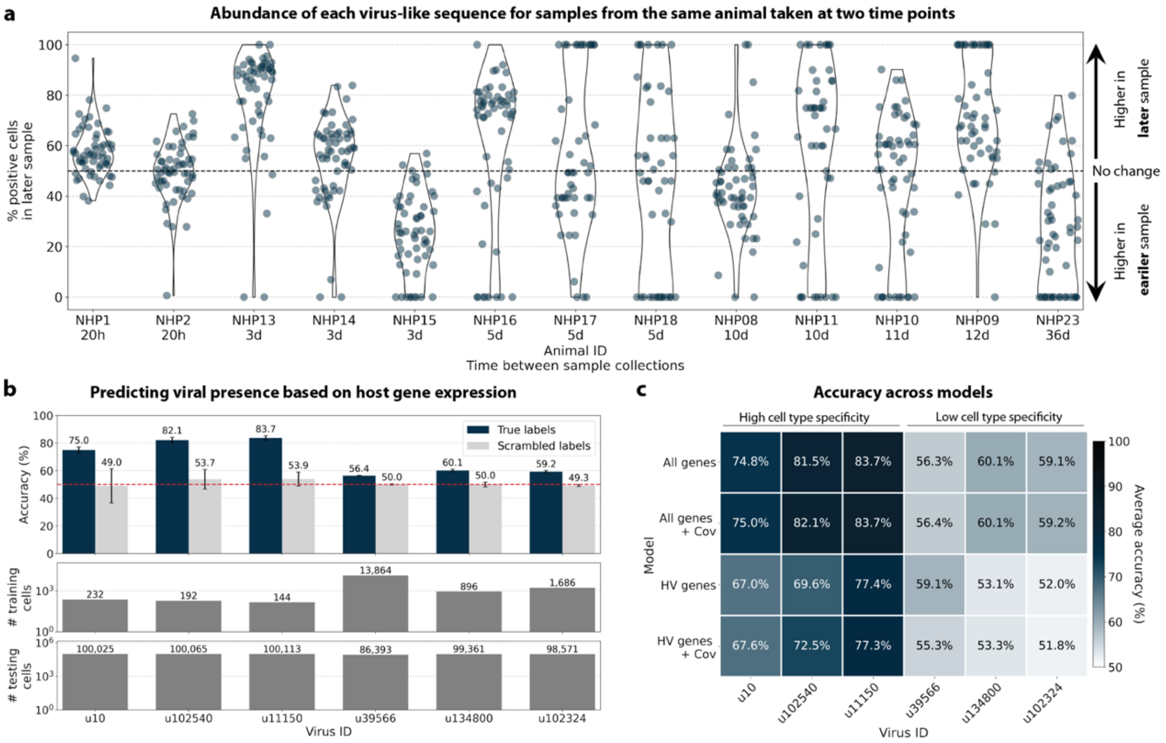


Fig. 3.8: a, Several animals included in the macaque PBMC dataset were sampled twice, at two different time points. Here, for each virus-like sequence, the percentage of positive cells occupied by the later time point is shown. The number of positive cells for each virus-like sequence was first normalized to the total number of cells in the sample. Only virus-like sequences for which at least one time point had positive counts were included for each animal. A percentage of 50% indicates that the number of positive cells for that virus-like sequence remained stable between the two time points. **b**, We trained logistic regression models to predict the presence of specific virus-like sequences based on host gene expression at single-cell resolution. The accuracy of the logistic regression model trained on all macaque genes with donor animal and EVD time point as covariates is shown for the known virus ZEBOV (u10) and five novel virus IDs. The presence of virus-like sequences that displayed high cell type specificity could be predicted with >70 % accuracy, while virus-like sequences with low cell type specificity could not be predicted above random chance (50 %, marked by the red dashed line). As a negative control, viral presence and absence labels were scrambled at random in the training data, causing the prediction accuracy to drop to random chance (50 %), as expected. The error bars indicate the standard deviation between models initialized with different random seeds. The bottom bar plots show the number of testing and training cells for each virus (also see Extended Data Fig. 3.8c). **c**, Heatmap of the prediction accuracy (averaged across models initialized with different random seeds) across all possible modeling combinations (training on all macaque genes versus only highly variable (HV) genes, and with or without covariates donor animal and EVD time point).

predicted better than random chance (50 %) (Fig. 3.8b and c, Extended Data Fig. 3.8a). As a negative control, we scrambled the binary virus count matrix used for model training, effectively randomizing the presence or absence of a virus in each cell. As expected, the prediction accuracies dropped to those expected at random (50 %) (Fig. 3.8b and c). We also confirmed that the different virus-like sequences with high prediction accuracy, including the known infection with ZEBOV (u10), were not present in the same cells (Extended Data Fig. 3.7b).

This learnable relationship between viral presence and host gene expression provides further evidence that reads from u10, the known infection with ZEBOV, as well as novel virus-like sequences u102540 and potentially u11150, originated from an ongoing viral infection and/or viral clearance which perturbed host gene expression at the single-cell level.

The only other virus-like sequence which displayed prediction accuracies > 70% was u202260 (Extended Data Fig. 3.8e). This was surprising, as u202260 was categorized as shared between macaque and MDCK cells and was also present in the blank sequencing libraries (Fig. 3.6b, Extended Data Fig. 3.9c), indicating that it likely originated from laboratory reagent contamination. However, although its prediction accuracies were relatively high, the gene weight correlation between different models was low for u202260 (Extended Data Fig. 3.8b) and the standard deviation of gene weights within the same model generated with different random seeds was comparatively high (Extended Data Fig. 3.9a), indicating that genes were weighted differently between models and seeds for u202260. This suggests that a shared feature across genes, such as cell health or sequencing depth, was learned rather than the expression of specific genes.

To explore virus-induced host gene expression, we identified macaque genes with the largest predictive power and smallest variation (across models initialized with different random seeds), for the regression models trained on highly variable genes with the donor animal and time point as covariates (Extended Data Fig. 3.9a). Approximately one third of the macaque Ensembl IDs did not have annotated gene names, which is a common problem for genomes from non-model organisms. We used gget[45] to translate annotated Ensembl IDs to gene symbols and to perform an enrichment analysis on the returned gene symbols using Enrichr[71] against the 2023 Gene Ontology (GO) Biological Processes database[72]. The highly weighted genes for u10 (ZEBOV) returned significant enrichment results for several virus-associated GO terms including 'Negative Regulation Of Viral Entry Into Host Cell (GO:0046597)', 'Negative Regulation Of Viral Life Cycle (GO:1903901)', and 'Regulation Of Viral Entry Into Host Cell (GO:0046596)', validating our approach for the identification of genes associated with a virus-related host gene response. Similarly, the enrichment analysis of highly weighted genes for the novel virus ID u11150 mapped to 'Receptor-Mediated Endocytosis Of Virus By Host Cell (GO:0019065).' For virus ID u102540, several highly ranked GO terms were indicative of an inflammatory response, such as 'Positive Regulation Of Type II Interferon Production (GO:0032729)' and 'Positive Regulation Of Cytokine Production (GO:0001819)'. Several predictive genes were associated with the positive regulation of cytokine production and modulation of inflammation (e.g., FCN1 for u 10, MAPK11 for u11150, and CD14 for u102540). Overall, these results provide further evidence that the novel virus-like sequences u102540 and u11150 originated from an ongoing viral infection or clearance resulting in a host gene response.

Discussion

Our work provides a method for extracting a 'virome' modality from any bulk or single-cell RNA-seq data by leveraging a new method that maps and quantifies species-level viral

RdRP sequences against an amino acid reference. We built on the existing alignment software kallisto[44] and bustools[76] and expanded them for translated alignment by (reverse) translating both the amino acid reference and the nucleotide sequencing reads into a common, nonredundant comma-free code. While we validated kallisto translated search in combination with PalmDB for the identification of viral RNA, our novel workflow can be applied in combination with any amino acid reference. kallisto translated search permits the alignment of nucleotide sequencing data to any amino acid reference at single-cell resolution. For example, amino acid sequences of antimicrobial peptides[77] can be used as a reference to identify these peptides in bulk and single-cell RNA sequencing data. Moreover, amino acid transcriptomes of homologous species may be used as a reference for species with missing or incomplete reference genomes. In this case, operating in the amino acid space will increase similarity due to the robustness to single-nucleotide mutations.

We validated kallisto translated search in combination with PalmDB for the detection and identification of viral RNA in next-generation sequencing data at single-cell resolution. As we noted in the introduction, the number of viruses expected to cause human infectious disease is eclipsed by the comparatively few viruses with complete reference genomes and the even smaller number of viruses that have been detected in humans. It is important to monitor the presence of viruses in the human population, both to prevent pandemic outbreaks and to further understand the role of viruses in various diseases. We have shown that such monitoring and novel virus discovery can be performed using single-cell RNA-seq data. Moreover, our work provides a platform for characterizing omnipresent virus-like sequences associated with different environments, hosts, and laboratory reagents.

The virus count matrix, which is obtained using kallisto translated search in combination with PalmDB, is an entirely new modality that we have begun to explore in this paper. We found that this matrix is sparse with relatively low molecule counts per cell (Extended Data Fig. 3.3e). While using the highly conserved RdRP to identify viruses makes our workflow very efficient and is the key to being able to detect over 100,000 distinct viruses, RdRP RNA only makes up ~1 % of the total viral RNA present in the sequencing data analyzed here (Extended Data Fig. 3.2) resulting in the sparsity of the virus count matrix. We anticipate that this number varies between virus species and sequencing technology, making it difficult to define a general detection limit. To normalize this sparse and low-count matrix, we binarized the virus count matrix such that each cell was either positive or negative for each virus. Given the low counts, we expect that there is a high occurrence of false negatives in the virus count matrix while the confidence in positive cells is high. However, we have shown that relationships between viral presence and host gene signatures can be learned regardless.

A common problem in the identification of microbial sequences is the misidentification of host sequences as microbial. The PalmDB is not a curated database, and it is possible that some virus-like sequences in the PalmDB are not derived from viruses. In addition, differentiating between ongoing infections, reagent or sample contamination, cell-free RNA contamination, endogenous retroviruses, and widespread latent infections is a challenge. The kallisto translated search method computes both the virus count matrix and the host gene expression matrix at single-cell resolution, providing unique opportunities for parallel analysis of viral signatures and their effect on host gene expression. We describe different approaches to evaluate the nature of viral sequences identified by kallisto translated search, including taxonomic assignment of viruses based on the sOTU, analysis of viral tropism, extraction and BLAST alignment of raw sequencing reads identified as viral, and using a sample spike-in to categorize viruses into shared and sample-specific viruses. Moreover, we describe and evaluate different workflows to mask the host genome and/or transcriptome, allowing different levels of conservativeness and the quantification of sequencing reads that align to viral RdRPs as well as the host transcriptome. Notably, the efficacy of masking the host genome and/or transcriptome will depend on the quality and comprehensiveness of said genome/transcriptome. In this case, the majority of host sequences originated from rhesus macaque, which has a very comprehensive genome assembly[57]. Finally, we trained logistic regression models to predict viral presence at the single-cell level based on host gene expression, achieving high accuracy indicative of an ongoing viral infection or clearance. Our results show that it is beneficial to combine multiple of these approaches, which we validate and describe in detail, for the interpretation of the presence of virus-like sequences.

Focusing on the RdRP produces biases between virus species with varying life cycles, depending on the sequencing technology used. The genome of many negative-strand RNA (-ssRNA) is replicated as well as transcribed. Transcription produces short, often polyadenylated mRNA products which are captured and sequenced, including the RdRP. In contrast, the genome of many positive-strand RNA (+ssRNA) viruses undergoes replication, but not transcription. Instead, the genome is translated into polyproteins, which are subsequently cleaved. While +ssRNA virus genomes are often polyadenylated and hence are captured by polyA capture-dependent single-cell RNA sequencing technologies, sequencing ~100 bases from the polyA-tail using a poly(T) primer will not capture the RdRP if it is located too far from the polyA-tail (see the schematic overview of the SARS-CoV genome in Fig. 3,1). In this scenario, the RdRP of +ssRNA viruses will, however, be captured by bulk RNA sequencing and random hexamer primers in single-cell RNA sequencing (Extended Data Fig. 3.4c). Hence, sequencing using random hexamer primers overcomes the virus life cycle-dependent bias for single-cell technologies. Many novel sequencing technologies, including Parse Biosciences SPLiT-Seq[33], employ random hexamer primers to produce full-coverage sequencing and overcome biases introduced by poly(T) primers. We foresee that the use of random priming in sequencing will continue to increase. It is worth noting that, depending on the technology, intra-genomic sequences of +ssRNA viruses might be captured by poly(T) primers nonetheless due to mispriming. Even with random priming, many biases will remain. For example, any viral RNA that is not polyadenylated will not be captured efficiently by single-cell sequencing technologies that rely on polyA capture.

We hope that kallisto translated search will be widely implemented in the analysis of next-generation sequencing data to identify the presence of viral RNA, as well as inform the experimental design of research aiming to identify microbial reads from RNA sequencing data. We describe several experimental design choices that greatly impact the results of microbial read quantification, such as the sequencing primer design and sample spike-ins. The masking workflows described in this paper and the associated challenges are applicable to any metagenomics analysis beyond the identification of viral reads, and the workflows described here can be easily applied to nucleotide references, such as a 16S database for the characterization of the human gut microbiome[78].

Methods

Developing kallisto translated search and optimization for the identification of viral RNA

Building kallisto translated search and choosing a new 'genetic code'

To perform translated alignment, the nucleotide and amino acid sequences need to be translated into a shared 'language'. This might be achieved by translating nucleotides to amino acids or vice versa. Since kallisto encodes nucleotide characters in 2 bits (allowing a total of 4 distinct nucleotides to be encoded), encoding the 20 different amino acids resulting from translated nucleotide sequences was not feasible. Moreover, reverse translating the amino acid sequences to nucleotides would be intractable due to the redundancy in the genetic code leading to a combinatorial explosion in nucleotide sequences consistent with an amino acid reference. We therefore translated the nucleotide sequences and reverse translated the amino acid sequences using a fixed synthetic code designed to reduce spurious alignments. We explored two different codes for this translation: 1. Comma-free code and 2. A code that maximizes the Hamming distance between frequently occurring amino acids (Extended Data Fig. 3.10a). While maximizing Hamming distance is advantageous in terms of avoiding sequence multimapping (see next paragraph), a comma-free code prevents off-frame alignment since any k-mers formed by adjacent words will not be included in the dictionary. We found that the comma-free code recalls viral sequences equally well compared to maximizing the Hamming distance between amino acids (Extended Data Fig. 3.10b).

Optimization of PalmDB for the identification of viral reads in RNA sequencing data

Due to the occurrence of the ambiguous amino acid characters B, J, and Z, 62 out of 296,623 viral sequences were transformed into identical sequences after reverse translation to comma-free code. The identical sequences were merged and assigned a representative virus ID. Due to the high similarity between viral RdRP sequences, the loss of aligned sequences due to multimapping to several reference sequences was a major concern. Moreover, the necessity of reverse translating the amino acid sequences further decreases the Hamming distance between reference sequences by approximately 30 % (Extended Data Fig. 3.10d). To overcome this problem, we tried clustering the amino acid sequences based on 99 % similarity using the MMseqs2 algorithm[79]. This resulted in 6,518 clusters with high concordance of taxonomy labels between sequences in the same cluster (Extended Data Fig. 3.10e). However, although clusters were computed correctly based on their concordance with taxonomy, this resulted in 67.4 % of sOTUs not being detected anymore (compared to 3.3 % when using the complete index). As a result, we decided to group the sOTUs instead, treating virus IDs with the same taxonomy across all main taxonomic ranks like transcripts of the same gene (available here: https://tinyurl.com/4wd33rey). This retained the alignment percentage of the complete index while allowing highly accurate taxonomic assignment and minimal sequence loss to multimapping (Fig. 3.3b). It is noteworthy that the default kallisto k-mer length of 31 nucleotides equals only 10 amino acids. Given the architecture of the current kallisto version (0.50.1), which is optimized for 64-bit k-mers with each nucleotide occupying two bits, k cannot be set > 31. This will change in future versions.

Validation and benchmarks

Visualization of the Kraken2 and kallisto translated search alignments of ZEBOV sequences

Kotliar *et al.*[37] performed single-cell RNA sequencing of PBMC samples from rhesus macaques after infection with ZEBOV (further described below). A subset of the data obtained by Kotliar *et al.* at 8 days post-infection with ZEBOV was used to visualize the identification of RdRP sequences using kallisto (v0.50.1) translated search. The first 100,000,000 raw sequencing reads from the GSE158390 library SRR12698539 were aligned to the ZEBOV reference genome (NC_002549.1) using Kraken2 v2.1.2 and to the optimized PalmDB using kallisto translated search. Aligned reads from both workflows were extracted and realigned to the ZEBOV genome using bowtie2[39] v2.2.5 and SAMtools[40] v1.6 as previously described[80]. The visualization shown in Extended Data Fig. 3.2 was generated from the resulting sorted bam files with the NCBI Genome Workbench[41].

Testing robustness to mutation

676 *Zaire ebolavirus* (ZEBOV) RdRP sequences were identified by aligning the first 100,000,000 raw sequencing reads from the GSE158390 library SRR12698539 to the optimized PalmDB using kallisto translated search. Mutation-Simulator[48] (v3.0.1) was used to add random single nucleotide base substitutions to the RdRP sequences at increasing mutation rates. We performed 10 rounds of simulated mutations per mutation rate. The sequences were subsequently aligned using kallisto translated search against the complete PalmDB, Kraken2 translated search against the RdRP amino acid sequence of ZEBOV with a manually adjusted NCBI Taxonomy ID to allow compatibility with Kraken2, and kallisto standard workflow against the complete ZEBOV nucleotide genome (GCA_000848505.1). We subsequently calculated the recall percentage over all 676 sequences. For kallisto translated search, the recall percentage was calculated based on species-level taxonomic assignment. Since the other two methods were only given the target virus sequence as a reference and did not have to distinguish between different viruses, their recall percentage was calculated based on all aligned sequences. The recall percentage over all 676 sequences for the 10 rounds at each mutation rate is shown in Fig. 3.2c. Extended Data Fig. 3.4b shows the precision with which kallisto translated search identified the correct virus versus other taxonomies at each mutation rate. The recall and precision at mutation rates > 0 were fitted with an inverse sigmoid function using non-linear least squares using the scipy.optimize.curve_fit function (scipy v1.11.1).

Alignment and quantification of viral counts in validation datasets

The sequencing reads for each library used in the validation (Fig. 3.3a) were aligned with kallisto translated search against the PalmDB index D-listed with the corresponding host genome and transcriptome. The hosts were (i) human (GRCh38 Ensembl version 109) for GSE150316, GSM4548303 and the SARS-CoV-2 saliva, nasal, and throat samples, (ii) mouse (GRCm39 Ensembl version 109) for GSM5974202, and (iii) rhesus macaque (Mmul_10 Ensembl version 109) and dog (ROS_Cfam_1.0 Ensembl version 109) for GSE158390. Count matrices were generated with bustools (v0.43.1). Fig. 3.3a shows the total raw counts obtained for each target virus species. RT-qPCR and RNA-ISH counts were reproduced from the original publications.

Validating the alignment of nucleotide sequences to an amino acid reference and assessing the accuracy of the taxonomic assignment

To validate the mapping of nucleotide sequences to an amino acid reference with kallisto translated search and assess the accuracy of the taxonomic assignment, we reverse translated all amino acid sequences in the PalmDB using the 'standard' genetic code from the biopython[81] (v1.79) Bio.Data.CodonTable module and DnaChisel[82] (v3.2.10) (with a slight adaptation to allow the ambiguous amino acids 'X', 'B', 'J', and 'Z' occurring in the PalmDB, which was later implemented in DnaChisel v3.2.11). A unique synthetic 'cell barcode' was generated for each resulting nucleotide sequence, and the sequences were aligned to the optimized amino acid PalmDB with kallisto translated search, keeping track of each sequence separately as if they were an individual cell. The synthetic barcodes allowed subsequent analysis of the alignment result for each individual sequence, and the accuracy of the obtained taxonomy based on the virus ID to sOTU mapping provided by PalmDB is shown in Fig. 3.3b. For each sequence, we differentiated between 'correct' or 'incorrect' taxonomic assignment, or, if the sequence did not return any results, whether it was 'multimapped' (the sequence aligned to multiple targets in the reference and could not unambiguously be assigned to one) or 'not aligned' (the sequence was not aligned), at each taxonomic rank.

Analysis of macaque PBMC data

Kotliar *et al.*[37] performed single-cell RNA sequencing of PBMC samples from 19 rhesus macaques at different time points during Ebola virus disease (EVD) after infection with ZEBOV (EBOV/Kikwit; GenBank accession MG572235.1; *Filoviridae: Zaire ebolavirus*) using Seq-Well[74] with the S3 protocol[75]. A subset of PBMC samples were spiked with Madin-Darby canine kidney (MDCK) cells. The data is available at GSE158390, and we obtained the raw sequencing data from the European Nucleotide Archive using FTP download links and ffq (v0.3.0)[83]. The data is split into 106 datasets containing 30,594,130,037 reads in total.

Alignment to the host transcriptome

The rhesus macaque Mmul_10 and domestic dog ROS_Cfam_1.0 genomes were retrieved from Ensembl version 109. The reference index was built using both genomes and the kb-python (v0.28.0 with kallisto v0.50.1 and bustools v0.43.1) ref command to create a combined index containing the transcriptome of both species. We quantified the gene expression in each of the 106 datasets using the standard kallisto-bustools workflow[13] with the ‘batch’ and ‘batch-barcodes’ arguments to process all files simultaneously while keeping track of each batch, and with the ‘x’-string ‘0,0,12:0,12,20:1,0,0’ to match the Seq-Well technology. Since the Seq-Well technology does not provide a barcode on-list, we generated a barcode on-list using the ‘bustools allowlist’ command, requiring each barcode to occur at least 1,000 times. We subsequently corrected the cell barcodes using the generated on-list and computed the count matrix using the ‘bustools count’ function.

Host cell quality control, filtering, and separation of macaque and MDCK cells
The count matrix generated by bustools was converted to h5ad using kb_python.utils.kb_utils and read into Python using anndata v0.8.0. Metadata such as donor animal, the presence of an MDCK spike-in, and time point were added to the AnnData object from the SRR library metadata provided by Kotliar *et al.*[37]. The cell barcodes were filtered based on a minimum number of UMI counts of 125 obtained from the knee plot of sorted total UMI counts per cell (Extended Data Fig. 3.3a), resulting in a mean UMI count of 1,401 after filtering. The cells were further filtered based on a maximum percentage of mitochondrial genes of 10 %, based on a combination of macaque and dog mitochondrial genes facilitated by Scanpy[84] (v1.9.3) and gget[45] (v0.28.0). Cells were categorized as macaque if a maximum of 10 % of their UMIs originated from dog genes and vice versa (Extended Data Fig. 3.3b). Macaque and MDCK cells were normalized separately using log(CP10k + 1) with Scanpy’s normalize_total defaults of target sum 10,000 and log1p.

Macaque cell clustering and cell type assignment
The macaque gene count matrix was transformed by PCA to 50 dimensions applied using the log-normalized counts filtered for highly variable genes using Scanpy’s highly_variable_genes. Next, we computed nearest neighbors and conducted Leiden clustering[58] using Scanpy, resulting in 19 Leiden clusters. We found that EVD time points were highly concordant across sequencing libraries, suggesting the lack of a batch effect (Fig. 3.7a, also see GitHub code repository). Each cluster was manually annotated with a cell type based on the expression of previously established marker genes[37] (Extended Data Fig. 3.3d). Cluster ‘Undefined 1’ was omitted because it only contained 12 cells. Gene names and descriptions for Ensembl IDs without annotations were obtained using gget[45].

Virus alignment with different masking options
For each masking option, we quantified the gene expression in each of the 106 datasets from GSE158390 using kallisto with the ‘batch’ and ‘batch-barcodes’ arguments to process all files simultaneously while keeping track of each batch and with the ‘x’-string ‘0,0,12:0,12,20:1,0,0’ to match the Seq-Well technology. kallisto translated search was initiated in the ‘kallisto index’ and ‘kallisto bus’ commands by adding the ‘—aa’ flag. Following the alignment to PalmDB with any of the masking options, cell barcodes were corrected using the barcode on-list generated during the alignment to the host as described above.

- No mask
The raw sequencing reads were aligned to the optimized PalmDB reference files (see 'Optimization of PalmDB' above) using kallisto translated search.

- D-list genome + transcriptome
The raw sequencing reads were aligned to the optimized PalmDB reference using kallisto translated search with the added argument 'd-list', which was passed the concatenated macaque genome and transcriptome (Mmul_10 Ensembl version 109), and dog genome and transcriptome (ROS_Cfam_1.0 Ensembl version 109). For D-list masking options including only the genomes or transcriptomes (Extended Data Fig. 3.4a), only the genome or transcriptome files from both species were concatenated and passed to the 'd-list' argument, respectively.

- D-list genome + transcriptome + ambiguous reads filtered
This workflow was performed as described above for the 'D-list genome + transcriptome' with an unreleased version of kallisto where ambiguous reads in the D-list will be thrown out as host instead of being assigned to virus (Extended Data Fig. 3.4a). We explored this option to investigate the effect of ambiguous reads during D-list masking. However, we found that the alignment results did not notably differ from the masking option 'D-list genome + transcriptome' (Fig. 3.4a and Extended Data Fig. 3.5).

- Host read capture with kallisto
The raw sequencing reads were aligned to the combined macaque and dog reference index generated during the alignment to host with 'kallisto bus' with the added '-n' flag. The '-n' flag keeps track of the read line number of each aligned read; the line numbers are added to the resulting BUS file. The raw sequencing reads were also aligned to the modified PalmDB with kallisto translated search with the added '-n' flag to obtain all reads that map to viral RdRPs. Subsequently, the BUS file returned by kallisto translated search was split into reads that only aligned to viral RdRPs and reads that also aligned to host based on the read line numbers in the BUS files. This step was performed using 'bustools capture' to, first, obtain all reads that belonged to a single batch file (of the 106 dataset files), and, second, capture all reads that also aligned to host.

- Host read capture with kallisto + D-list genome + transcriptome
Host reads were captured with kallisto as described above under 'Host read capture with kallisto'. However, during the alignment of the raw sequencing reads to PalmDB with the '-n' flag, we also used the 'd-list' flag to mask the host genomes and transcriptomes as described above under 'D-list genome + transcriptome'.

- Prior alignment to host with bwa
bwa[54] v0.7.17 was installed from source. The 'bwa index' command was used to generate a bwa index from the concatenated macaque and dog genomes (Mmul_10 and ROS_Cfam_1.0 from Ensembl v109). The raw sequencing reads were subsequently aligned to the bwa index using the 'bwa mem' command, aligning each file separately. For each FASTQ file, the names of all unmapped reads were extracted using 'samtools view' (SAMtools[40] v1.6), and a new FASTQ file including only unmapped sequences was

generated using the 'seqtk subseq' command (v1.4; https://github.com/lh3/seqtk). The resulting FASTQ files containing the sequencing reads that did not map to the host genomes were aligned to the optimized PalmDB reference files using kallisto translated search.

Extraction and BLAST alignment of viral reads

Randomly selected sequencing reads from three libraries that included reads that mapped to the viruses of interest were aligned to the optimized PalmDB with kallisto translated search including the '-n' flag, without any host read masking. Reads that mapped to the viruses of interest were subsequently captured and extracted from the raw sequencing FASTQ files using 'bustools capture' and 'bustools extract'.

BLAST+[55,56] v2.14.1 was installed from source and the BLAST nt database was downloaded using the update_blastdb.pl command. 10 reads were randomly chosen for each target virus for each library and were BLASTed against the nt database using the blastn algorithm. Sequences that aligned to the polyA tail were recognized by the occurrence of 'AAAAAAAAAAAA' or 'TTTTTTTTTTTT' in the aligned part of the subject or query sequences and removed from the results. BLAST results were subsequently plotted using pyCirclize.Circos (v1.0.0; https://github.com/moshi4/pyCirclize).

Virus quality control

The viral count matrix generated using the 'Host read capture with kallisto + D-list genome + transcriptome' masking workflow was converted to h5ad using kb_python.utils.kb_utils and read into Python using anndata v0.8.0. Metadata such as donor animal, the presence of an MDCK spike-in, and time point were added to the AnnData object from the SRR library metadata provided by Kotliar *et al.*[37]. For each cell, the host species and cell type were added from the host matrices generated as described above. The virus count matrix was subsequently binarized, such that for each cell, each virus was either present or absent. The viruses were thresholded to viruses that were observed in ≥ 0.05 % of cells in either species.

Virus categorization into shared, 'macaque only', and 'MDCK only' viruses

For each virus ID, the virus was defined as 'shared' if the fold change between the fraction of positive macaque cells and the fraction of positive MDCK cells was less than or equal to 2. Viruses were assigned the category 'macaque only' if the virus was seen in ≥ 0.05 % of macaque cells and ≤ 7 MDCK cells, and vice versa for the category 'MDCK only'. These thresholds were defined based on the percentages of positive cells observed for each virus in each species, as shown in Fig. 3.6b.

Generation of the Krona plot

KronaTools[62] v2.8.1 was installed from source. We generated a data frame containing the total numbers of positive cells for each sOTU seen in ≥ 0.05 % of macaque cells for each animal and time point (including only cells that passed host cell quality control). The ktImportText tool was used to generate a Krona plot HTML file from a text file generated from this data frame.

Logistic regression models to predict viral presence based on host gene expression

Logistic regression models the log odds of an event as a weighted linear combination of some predictor variables. That is, the natural log of the ratio of the probability p that an event occurs to the probability that it does not occur is modeled:

$$ln\left(\frac{p}{1-p}\right) = \sum_{i=1}^{N} \beta_i x_i + \beta_0 ,$$

where each x_i is a predictor variable with corresponding weight β_i and β_0 is an intercept. Here, p is the probability of viral presence or absence in a given cell, predicted based on a linear combination of normalized host gene count values (denoted as x with a total of G modeled genes). Viral presence or absence is modeled for a single virus at a time. To control for covariates, we also included animal identifier (denoted as y with a total of A animals) and time point (denoted as z with a total of T time points), which were one-hot encoded for fits:

$$ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{g\in G} \beta_g^G x_g + \sum_{a\in A} \beta_a^A y_a + \sum_{t\in T} \beta_t^T z_t$$

The magnitude of the weight value for each predictor variable corresponds to that variable's influence on event probability, with large positive weights increasing the probability and large negative weights decreasing the probability of the event. Thus, for our purposes, an analysis of gene weights suggests which genes are likely to correlate with viral infection. For models parameterized by highly variable (HV) genes, the host (macaque) matrix was subset to highly variable genes as defined above. To reduce the occurrence of false negative viral counts, the logistic regression models were trained using the viral count matrix obtained without any masking of the host genes. However, the models were trained for viruses that were filtered based on the more conservative masking options ('macaque only' and 'shared' viruses). To further reduce the occurrence of false negative viral counts, we filtered the virus and host matrices to include only the top 50 % of cells according to the sum of raw host reads per cell before training the models. This was done to reduce the effects introduced by varying sequencing depths. For example, cells with a lower sequencing depth will have a higher likelihood of a false negative viral count.

For viruses with more virus-negative than virus-positive cells, half of the virus-positive cells and an equal number of virus-negative cells were randomly selected to train the logistic regression models. For viruses with more virus-positive than virus-negative cells, half of the virus-negative cells and an equal number of virus-positive cells were randomly selected for training. In both cases, the remaining cells were held out for testing the performance of trained models. Given the cell type specificity of the viruses whose presence could be predicted with high accuracy, we wanted to confirm that we were not simply predicting cell type. To this end, virus-negative training cells were selected to be of the same cell types as virus-positive cells (Extended Data Fig. 3.8d). The number of training and testing cells for each virus are shown in Extended Data Fig. 3.8c.

For models that included covariates, donor animal and EVD time point were one-hot encoded and appended to the gene expression training matrix. All models included an intercept. Models were trained with L2 weight regularization using the sklearn.linear_model.LogisticRegression (sklearn v1.0.1) classifier with a maximum of 100 iterations to predict the probability of viral presence at single-cell resolution. Virus-positive cells were assigned class label 1, and virus-negative cells were assigned class label 0. All four possible combinations of two modeling choices (highly variable versus all genes, and covariates versus no covariates) were tested, and the results are shown in Fig. 3.8c. Accuracy, specificity, and sensitivity were calculated for each model on the held-out testing cells (Extended Data Fig. 3.8a). A negative control where labels of viral presence and absence for each virus were randomly scrambled in the training data was included in the modeling experiments. For the scrambled labels, the original ratio of virus-positive to virus-negative cells was maintained.

All results were averaged across models generated using six different random seeds for parameter optimization and random selection of cells for training and testing.

Enrichment analysis of predictive genes

Of the top 50 highly variable macaque genes with the largest positive average weights in the regression model we selected those for which the standard deviation of the weights was less than half than the lowest weight. Here, we used the model trained on highly variable genes with covariates donor animal and time point. The gene weight distributions and thresholds are shown in Extended Data Fig. 3.9a. Approximately one third of the macaque Ensembl IDs did not have annotated gene names. We used gget[45] to translate annotated Ensembl IDs to gene symbols and to perform enrichment analysis on the returned gene symbols using Enrichr[71] against the 2023 Gene Ontology (GO) Biological Processes database ('GO_Biological_Process_2023')[72]. The reported P values were corrected with the Benjamini-Hochberg method[73].

Data availability

Sample	Methods	Data shown in	DOI	GEO accession
Lung autopsy samples from COVID-19 patients	Bulk RNA sequencing; RNA-ISH	Fig. 3.3a	https://doi.org/10.1038/s41467-020-20139-7	GSE150316
Self-collected saliva, anterior nares swab, and oropharyngeal swab samples from individuals enrolled in a COVID-19 household transmission study	Bulk RNA sequencing with viral surveillance panel enrichment (Illumina Cat. 20040536 and 20088154); RT-qPCR	Fig. 3.3a	https://doi.org/10.1128/spectrum.03873-22 https://doi.org/10.1093/pnasnexus/pgad033	Raw sequencing data is not publicly available per participant privacy practices
SARS-CoV-2 infected human iPSC derived cardiomyocytes	SMART-Seq V4	Fig. 3.3a	https://doi.org/10.1016/j.xcrm.2020.100052	GSM4548303
Blood samples from rhesus macaques infected with *Zaire ebolavirus*	Seq-Well S^3; RT-qPCR	Fig. 3.2-8; Extended Data Fig. 3.1-9	https://doi.org/10.1016/j.cell.2020.10.002	GSE158390
Lungs from APOE knock-in mice infected with SARS-CoV-2	SPLiT-Seq (Parse Biosciences)	Extended Data Fig. 3.4c	https://doi.org/10.1038/s41586-022-05344-2	GSM5974202

Table 3.1: Availability of data analyzed in this paper.

File	Description	Category
viral sequences in laboratory reagents.h5ad	Count matrix containing virus-like sequences found in sequencing libraries comprised of only sterile water and laboratory reagents	Alignment of 'blank' sequencing libraries to the PalmDB
host alignment results.zip	Raw alignment results obtained by kallisto after alignment to the macaque and dog (to account for the MDCK spike-in) transcriptomes	Alignment of the macaque PBMC data[37] to the host transcriptome(s)
host QC.h5ad	Filtered count matrix containing all host cells	
canis QC norm leiden.h5ad	Filtered and clustered count matrix containing MDCK cells	
macaque QC norm leiden.h5ad	Filtered and clustered count matrix containing macaque cells	
macaque QC norm leiden celltypes.h5ad	Filtered and clustered count matrix containing macaque cells with cell type assignments	
virus no mask alignment results.zip	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB without masking host sequences	Alignment of the macaque PBMC data[37] to the PalmDB for the detection of viral RNA with different workflows for the masking of host genome(s) and transcriptome(s)
virus no mask.h5ad	Count matrix obtained through the alignment above with added metadata	
virus dlist cdna alignment results.zip	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB while masking host transcriptome(s) using the D-list	
virus dlist cdna.h5ad	Count matrix obtained through the alignment above with added metadata	
virus dlist dna alignment results.zip	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB while masking host genome(s) using the D-list	
virus dlist dna.h5ad	Count matrix obtained through the alignment above with added metadata	
virus dlist cdna dna alignment results.zip	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB while masking host genome(s) and transcriptome(s) using the D-list	
virus dlist cdna dna.h5ad	Count matrix obtained through the alignment above with added metadata	
virus dlist cdna dna amb alignment results.zip	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB while masking host genome(s) and transcriptome(s) using the D-list + forcing ambiguous sequences to be discarded	
virus dlist cdna dna ambiguous.h5ad	Count matrix obtained through the alignment above with added metadata	
virus host capture alignment results.tar.gz	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB + reads that align to the host transcriptome(s) were captured	
virus host-captured.h5ad	Count matrix obtained through the alignment above with added metadata	
virus host capture dlist cdna dna alignment results.tar.gz	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB while masking host genome(s) and transcriptome(s) using the D-list + reads that align to the host transcriptome(s) were captured	
virus host-captured dlist cdna dna.h5ad	Count matrix obtained through the alignment above with added metadata	
bwa unmapped reads.tar.gz	Raw sequencing files obtained after removal of host sequences based on alignment with bwa	
virus bwa alignment results.zip	Raw alignment results obtained by kallisto translated search after alignment to the PalmDB after reads that align to the host genome(s) with bwa were removed	
virus bwa.h5ad	Count matrix obtained through the alignment above with added metadata	

models.zip	Logistic regression models to predict viral presence based on host gene expression	Logistic regression models
palmdb human dlist cdna dna.idx	Pre-computed PalmDB reference index with human genomic and transcriptomic sequences masked using D-list	Pre-computed references for future use with kallisto translated search
palmdb mouse dlist cdna dna.idx	Pre-computed PalmDB reference index with mouse genomic and transcriptomic sequences masked using D-list	

Table 3.2: Availability of data generated in this paper. The data is available on Caltech Data under the DOIs 10.22002/krqmp-5hy81 and 10.22002/k7xqw-88d74.

The PalmDB reference files optimized for use with kallisto translated search for the identification of viral sequences in bulk and single-cell RNA sequencing data are available here: https://tinyurl.com/4wd33rey.

The data generated in this paper is freely and publicly available on Caltech Data under the DOIs 10.22002/krqmp-5hy81 and 10.22002/k7xqw-88d74.

Code availability

The code used to generate all of the results and figures reported in this paper, starting from the raw sequencing reads, can be found here: https://github.com/pachterlab/LSCHWCP_2023. The code is organized by figure panel and provided in immediately executable Google Colab notebooks to maximize the reproducibility of the results and methods described in this manuscript.

Acknowledgments

We would like to thank Astrid Bryon for her advice and helpful discussions on the applicability of kallisto translated search for the detection of viral RNA. We would like to thank Aaron Lin, Lisa Hensley, Dylan Kotliar, and Pardis Sabeti for helping us understand and work with their data. We would like to thank Christina Middle, Morgan Roos, and Scott Kuersten from Illumina for generating the bulk RNA sequencing data included in Fig. 3.3a (second panel). The illustrations in Figures 1, 4, and 6 were created with BioRender.com.

Funding

LL was supported by funding from the Biology and Bioengineering Division at the California Institute of Technology. DKS was supported by funding from the UCLA-Caltech Medical Scientist Training Program (NIH NIGMS training grant T32 GM008042). MC was supported by the NSF GRFP under Grant No. 2139433. AVW was supported by NIH F30AI167524. AVW was supported in part by the Bill & Melinda Gates Foundation (INV-023124). Under the grant conditions of the Foundation, a Creative Commons Attribution 4.0 Generic License has already been assigned to the Author Accepted Manuscript version that might arise from this submission.

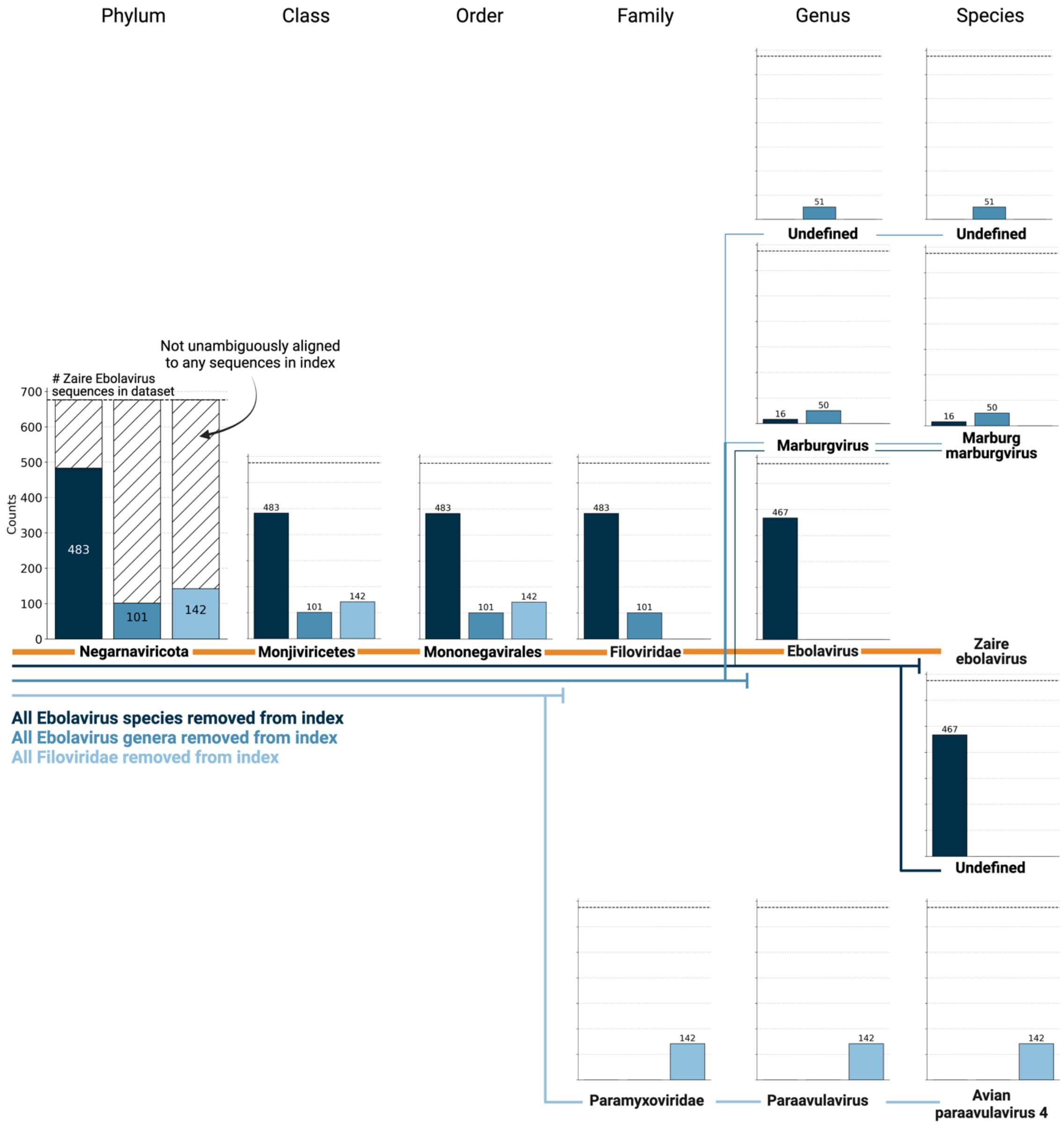
Ethics declarations

Competing Interests: LL, DKS, and LP are listed as inventors of a patent application relating to the work. The patent application was submitted through the Technology

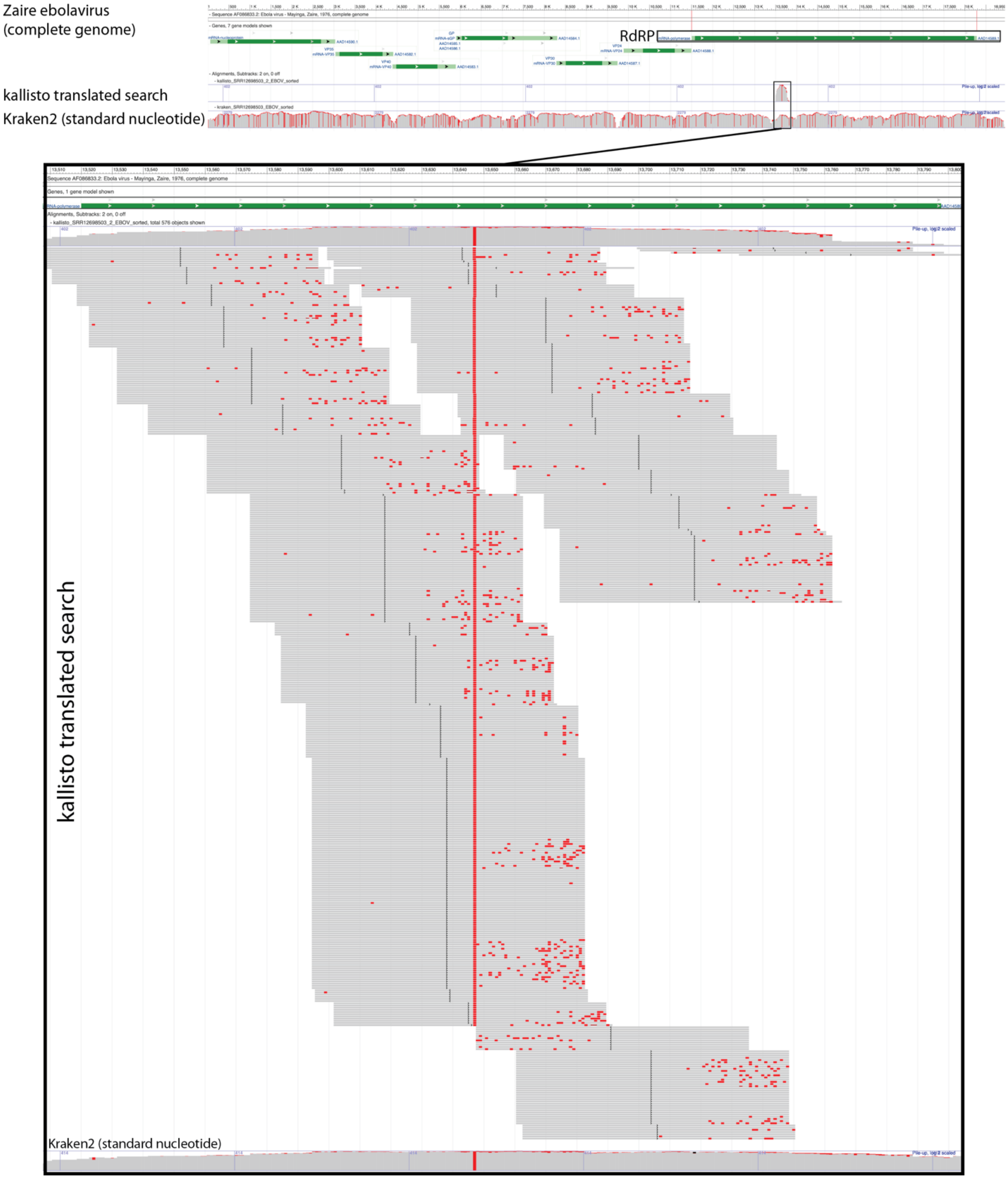
Transfer Office of the California Institute of Technology (Caltech), with Caltech being the patent applicant.

Virus ID	Phylum	Class	Order	Family	Genus	Species
u102540	*Pisuviricota*	*Pisoniviricetes*	*Nidovirales*	*Coronaviridae*	*Alphacoronavirus*	.
u10	*Negarnaviricota*	*Monjiviricetes*	*Mononegavirales*	*Filoviridae*	*Ebolavirus*	*Zaire ebolavirus*
u10240	*Pisuviricota*	*Pisoniviricetes*	*Nidovirales*	*Arteriviridae*	.	.
u1001	*Negarnaviricota*	*Insthoviricetes*	*Articulavirales*	*Orthomyxoviridae*	*Alphainfluenzavirus*	*Influenza A virus*
u100291	*Negarnaviricota*	*Monjiviricetes*	*Mononegavirales*	.	.	.
u103829	*Negarnaviricota*	*Insthoviricetes*	*Articulavirales*	*Orthomyxoviridae*	.	.
u110641	*Duplornaviricota*	*Resentoviricetes*	*Reovirales*	*Reoviridae*	.	.
u101227	*Pisuviricota*	*Pisoniviricetes*	*Picornavirales*	*Picornaviridae*	.	.
u100188	*Kitrinoviricota*	*Alsuviricetes*	*Martellivirales*	*Closteroviridae*	.	.
u27694	*Peploviricota*	*Herviviricetes*	*Herpesvirales*	*Herpesviridae*	*Varicellovirus*	*Bubaline alphaherpesvirus 1*
u100245	.	.	.	*Fusariviridae*	.	.
u10015	*Duplornaviricota*	*Chrymotiviricetes*	*Ghabrivirales*	*Totiviridae*	.	.
u100733	*Negarnaviricota*	.	.	.	.	.
u100173	*Lenarviricota*	*Miaviricetes*	*Ourlivirales*	*Botourmiaviridae*	*Ourmiavirus*	.
u100196	*Negarnaviricota*	*Monjiviricetes*	.	.	.	.
u100599	*Negarnaviricota*	*Ellioviricetes*	.	.	.	.
u100644	*Lenarviricota*	*Amabiliviricetes*	*Wolframvirales*	.	.	.
u100296	*Pisuviricota*	*Pisoniviricetes*	*Picornavirales*	*Dicistroviridae*	.	.
u100017	*Lenarviricota*	*Allassoviricetes*	*Levivirales*	*Leviviridae*	.	.
u100002	*Lenarviricota*	*Allassoviricetes*	*Levivirales*	.	.	.
u100012	*Lenarviricota*	*Allassoviricetes*	.	.	.	.
u100024	*Pisuviricota*	*Duplopiviricetes*	*Durnavirales*	*Picobirnaviridae*	.	.
u100048	*Lenarviricota*	*Amabiliviricetes*	.	.	.	.
u100302	*Negarnaviricota*	*Monjiviricetes*	*Mononegavirales*	*Rhabdoviridae*	.	.
u100074	*Lenarviricota*	*Howeltoviricetes*	*Cryppavirales*	.	.	.
u100289	*Negarnaviricota*	*Ellioviricetes*	*Bunyavirales*	.	.	.
u100026	*Pisuviricota*	*Duplopiviricetes*	*Durnavirales*	.	.	.
u100111	*Duplornaviricota*	*Chrymotiviricetes*	*Ghabrivirales*	.	.	.
u100139	*Kitrinoviricota*	*Alsuviricetes*	*Martellivirales*	.	.	.
u100154	*Pisuviricota*	*Duplopiviricetes*	*Durnavirales*	*Amalgaviridae*	.	.
u100251	*Pisuviricota*	*Duplopiviricetes*	.	.	.	.
u100177	*Kitrinoviricota*	*Tolucaviricetes*	*Tolivirales*	.	.	.
u100215	*Duplornaviricota*	*Chrymotiviricetes*	.	.	.	.
u100049	*Lenarviricota*	*Miaviricetes*	.	.	.	.
u100000	*Kitrinoviricota*	*Tolucaviricetes*	*Tolivirales*	*Tombusviridae*	.	.
u100001	*Lenarviricota*	*Howeltoviricetes*	*Cryppavirales*	*Mitoviridae*	.	.
u100007	*Lenarviricota*	.	.	.	.	.
u100004	*Lenarviricota*	*Miaviricetes*	*Ourlivirales*	.	.	.
u100011	*Lenarviricota*	*Howeltoviricetes*	.	.	.	.
u100093	*Pisuviricota*	*Duplopiviricetes*	*Durnavirales*	*Partitiviridae*	.	.
u100116	*Pisuviricota*	*Pisoniviricetes*	.	.	.	.
u100019	*Pisuviricota*	*Duplopiviricetes*	*Durnavirales*	*Picobirnaviridae*	*Picobirnavirus*	.
u100076	*Kitrinoviricota*	*Tolucaviricetes*	.	.	.	.
u100028	*Pisuviricota*	.	.	.	.	.
u100153	*Lenarviricota*	*Miaviricetes*	*Ourlivirales*	*Botourmiaviridae*	.	.
u100031	*Kitrinoviricota*	*Alsuviricetes*	.	.	.	.
u100145	*Pisuviricota*	*Pisoniviricetes*	*Sobelivirales*	.	.	.
u102324	*Pisuviricota*	*Pisoniviricetes*	*Picornavirales*	*Iflaviridae*	.	.

Extended Data Table 3.1: Virus ID to species-like operational taxonomic unit (sOTU) mapping for the most highly expressed viruses (in the same order as shown in Fig. 3.6d). Virus IDs that are further mentioned in the paper are marked in blue. Virus IDs not included in this list are of unknown taxonomy across all taxonomic ranks.



Extended Data Fig. 3.1: 676 ZEBOV RdRP sequences were identified by aligning a subset of 100,000,000 single-cell RNA sequencing reads of macaque PBMC samples obtained at 8 days post-infection with ZEBOV[37] to the optimized PalmDB using kallisto translated search. We subsequently aligned the sequences to PalmDB reference indices from which (i) all *Ebolavirus* species were removed (dark blue), (ii) all *Ebolavirus* genera were removed (medium blue), or (iii) all Filoviridae were removed (light blue). In each scenario, a subset of sequences aligned to the nearest remaining relative based on the main taxonomic rank, suggesting that kallisto translated search can detect the highly conserved RdRP of a large number of viral species, beyond the number of sequences in the PalmDB database, while still providing reliable sOTU-based taxonomic assignment of lower-rank taxonomies.


Zaire ebolavirus
(complete genome)
RdRP
kallisto translated search
Kraken2 (standard nucleotide)
kallisto translated search
Kraken2 (standard nucleotide)

Extended Data Fig. 3.2: Visualization of the identification of RdRP sequences with kallisto translated search. We selected a subset of 100,000,000 reads obtained using Seq-Well sequencing of macaque peripheral blood mononuclear cell (PBMC) samples obtained at 8 days post-infection with ZEBOV[37]. We aligned the reads to the PalmDB amino acid sequences with kallisto translated search. We also aligned the reads to the complete ZEBOV nucleotide genome using Kraken2 (standard nucleotide alignment)[27]. Aligned reads from both alignments were extracted and realigned to the ZEBOV genome using bowtie2[39], a BAM file was created using SAMtools[40], and the alignment was subsequently visualized NCBI Genome Workbench[41].

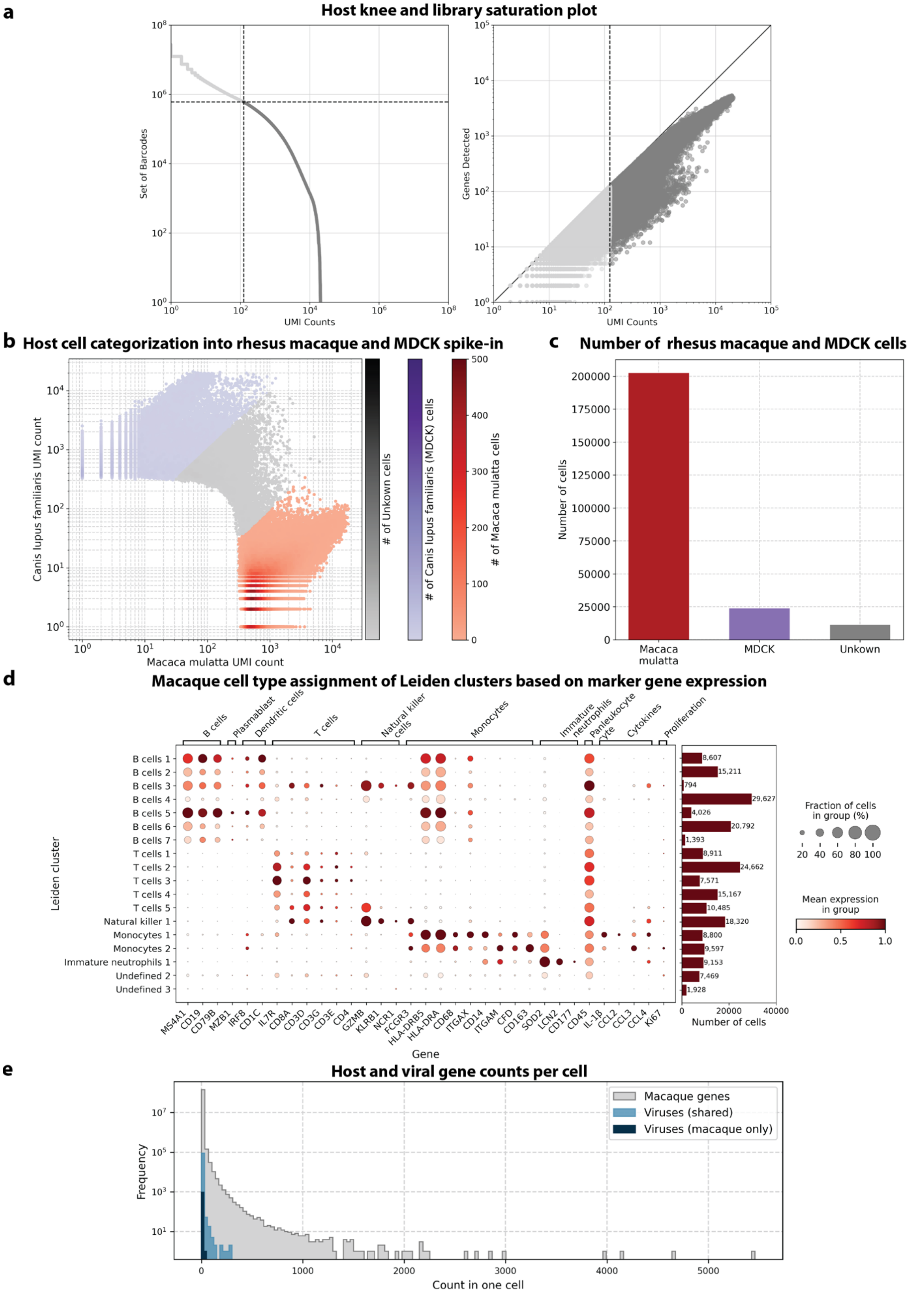


Figure legend on next page.

Extended Data Fig. 3.3: **a**, Knee plot of sorted total UMI counts per cell and library saturation plot of host (rhesus macaque and MDCK) cells sequenced by Kotliar *et al.*[37] **b,** Canis lupus (dog/MDCK) over Macaca mulatta (macaque) UMI count for each cell. Cells were categorized as macaque if a maximum of 10 % of their UMIs originated from dog genes and vice versa. **c**, The obtained numbers of macaque, dog (MDCK), and uncategorized cells after species separation. **d**, Mean expression of marker genes used for cell type assignment per macaque Leiden cluster. The barplot shows the number of cells in each cluster. Cluster 'Undefined 1' was omitted because it only contained 12 cells. **e**, Frequency of host and viral gene counts in individual cells.

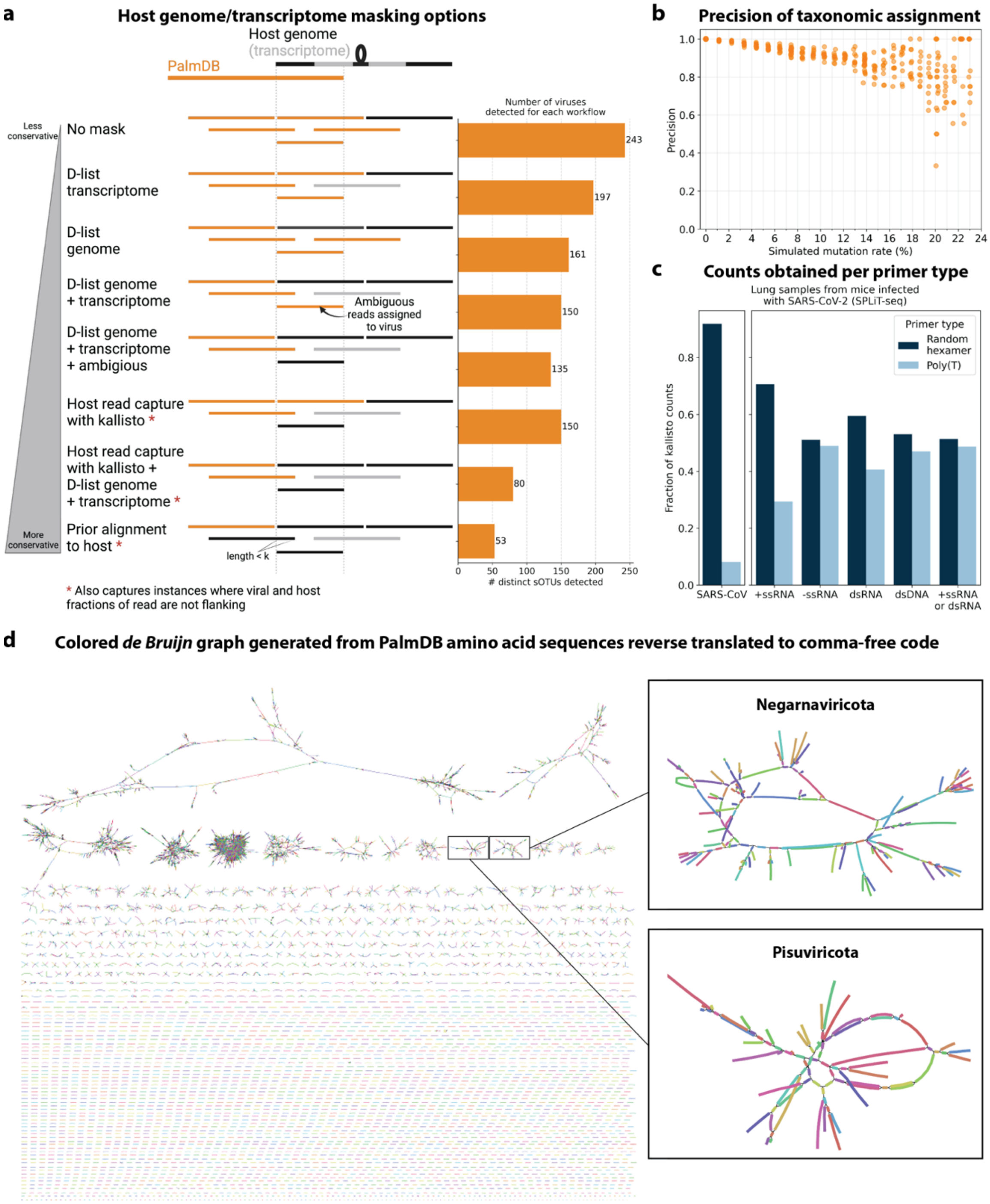
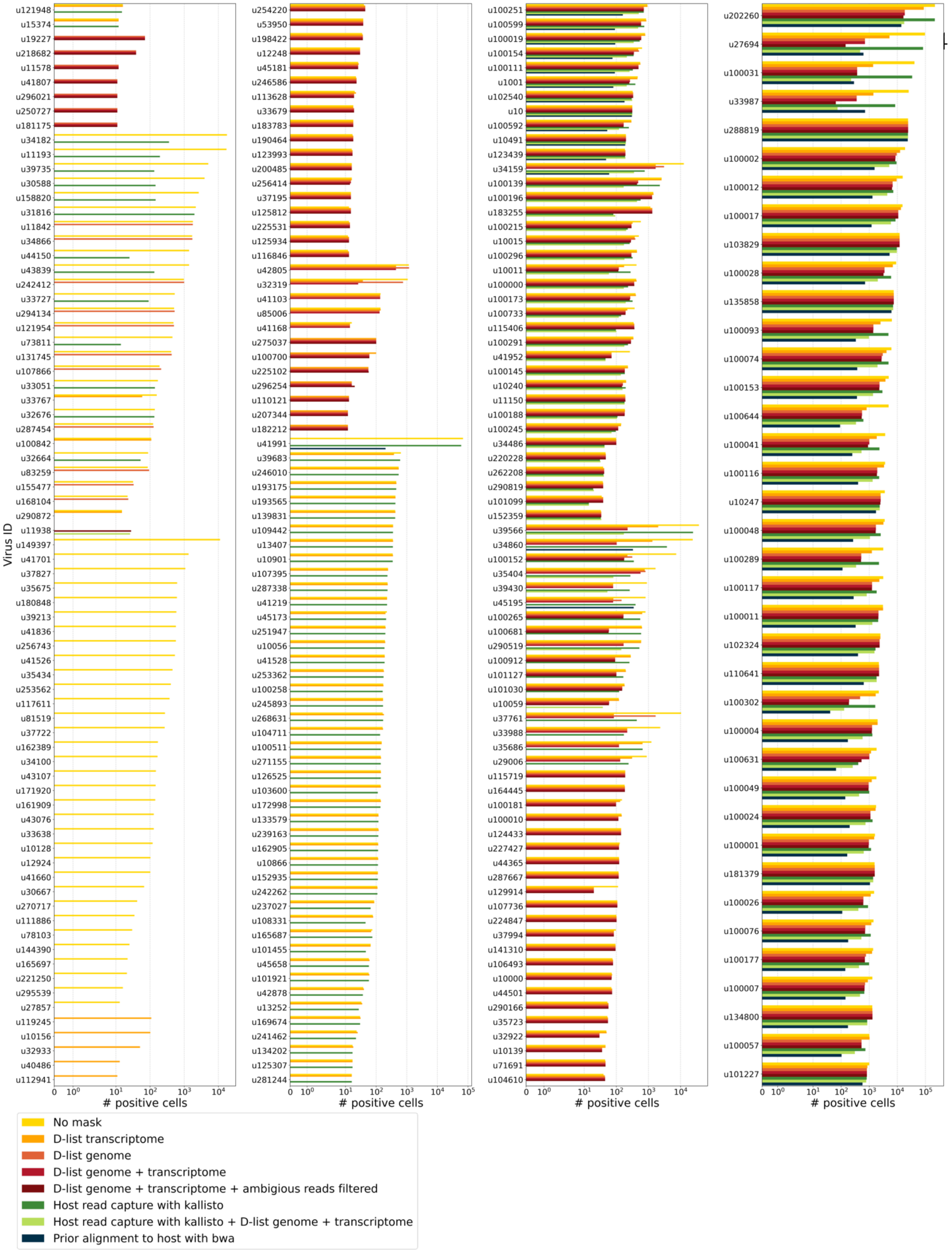
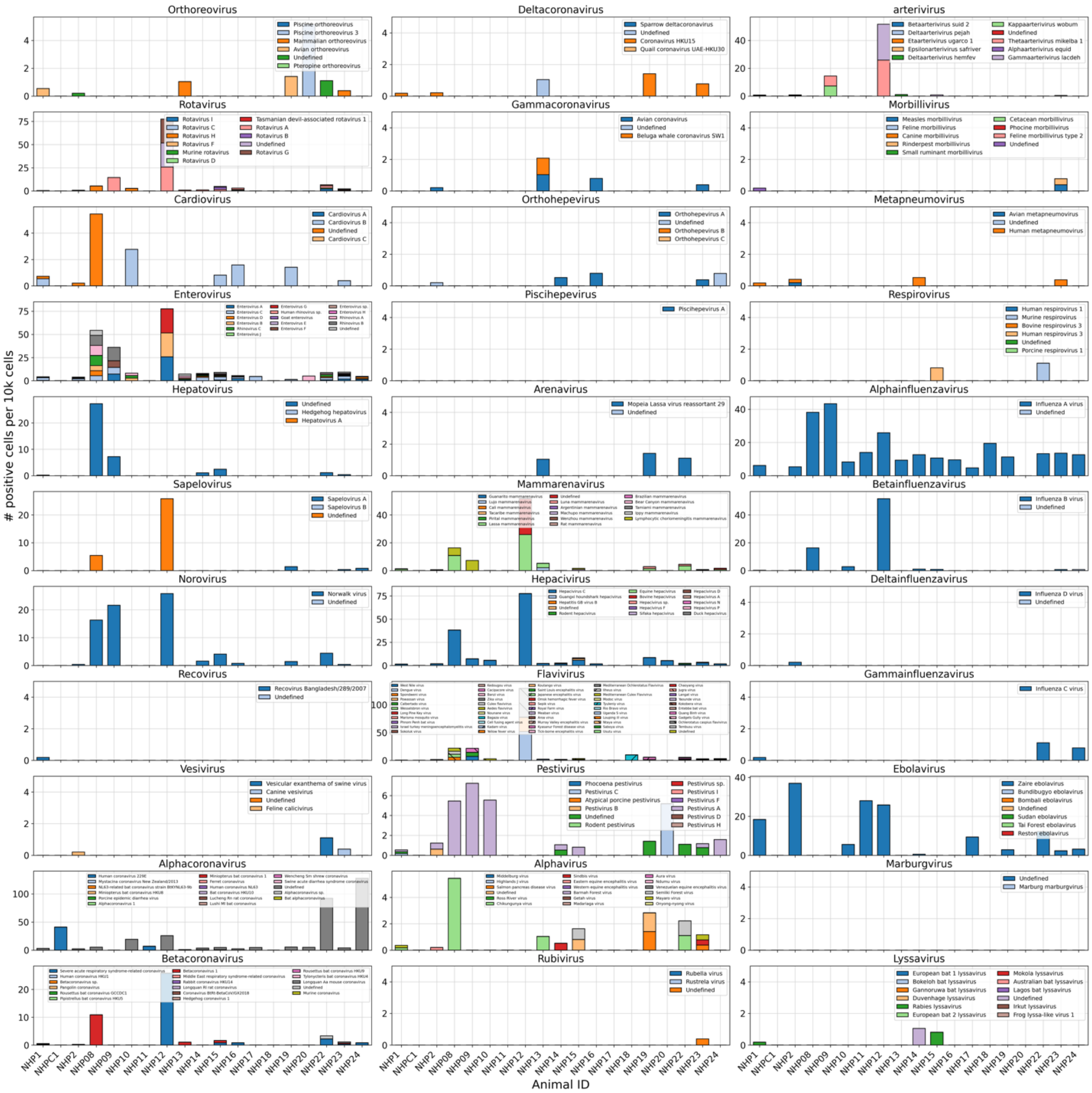


Figure legend on next page.

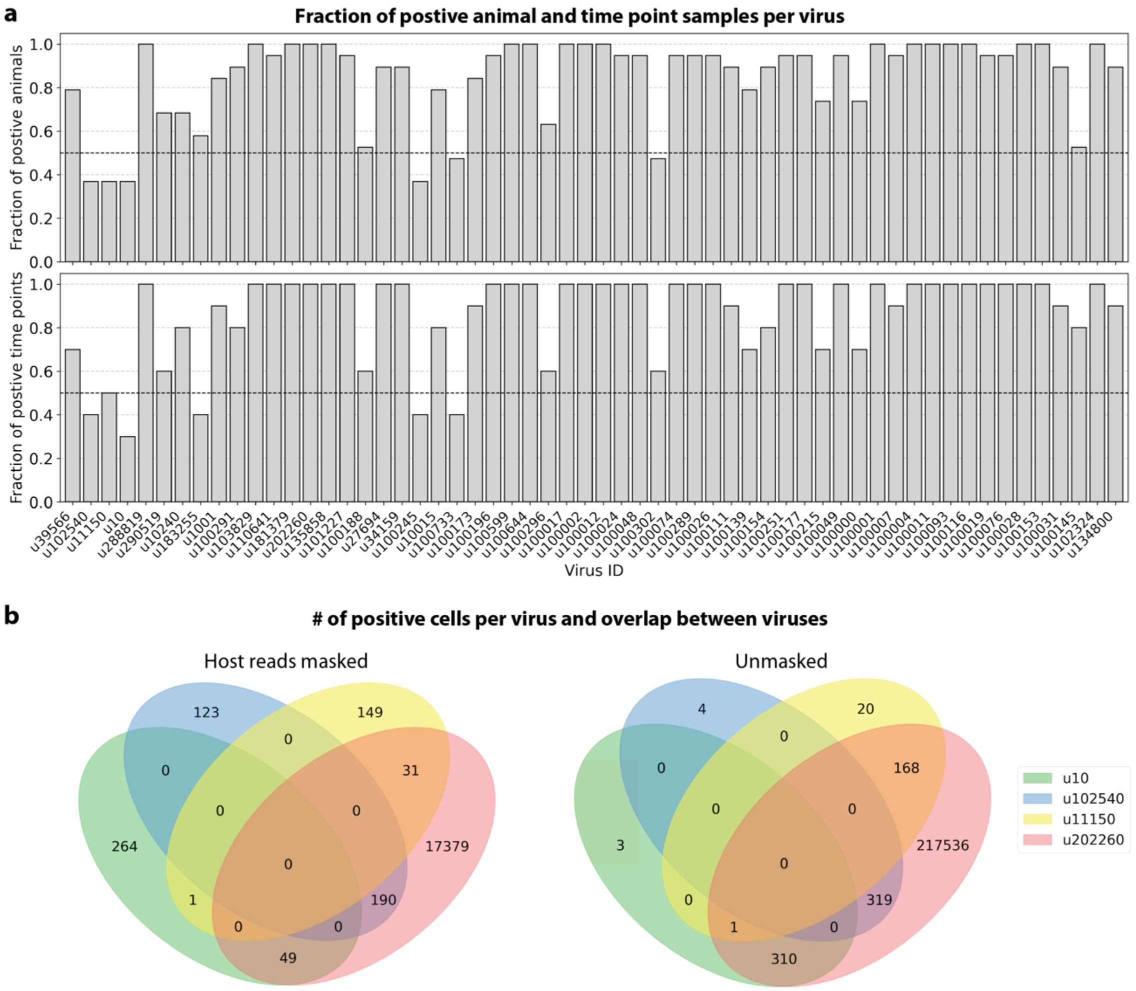
Extended Data Fig. 3.4: a, Schematic overview of different host masking options, extending the masking options shown in Fig. 3.4a. Reads that align to PalmDB and are considered viral are marked in orange, and reads that align to the host genome or transcriptome are marked in black or grey, respectively. The barplot shows the number of distinct sOTUs, defined by distinct virus IDs, observed in ≥ 0.05 % of cells for each workflow. **b**, Precision of species-level taxonomic assignment at increasing simulated mutation rates. Mutation-Simulator[48] was used to add random single nucleotide base substitutions to 676 ZEBOV RdRP sequences obtained by Seq-Well sequencing[37] at increasing mutation rates. We performed 10 simulations per mutation rate. The sequences were subsequently aligned using kallisto translated search against the complete PalmDB. The recall percentages at each mutation rate are shown in Fig. 3.2c. **c**, Fraction of counts obtained for the known viral infection (here, SARS-CoV-2) and per viral strandedness of other sOTUs per primer type. Lung samples from mice infected with SARS-CoV2 were sequenced with SPLiT-Seq[85] and aligned to PalmDB using kallisto translated search using the D-list to mask the host (here, mouse) genome. The plot shows the fraction of counts obtained for SARS-CoV as well as all sOTUs of different strandedness per primer type. **d**, The de Bruijn graph generated from the reverse translated PalmDB sequences in the kallisto translated search workflow, visualized and colored using Bandage v0.8.1[86].



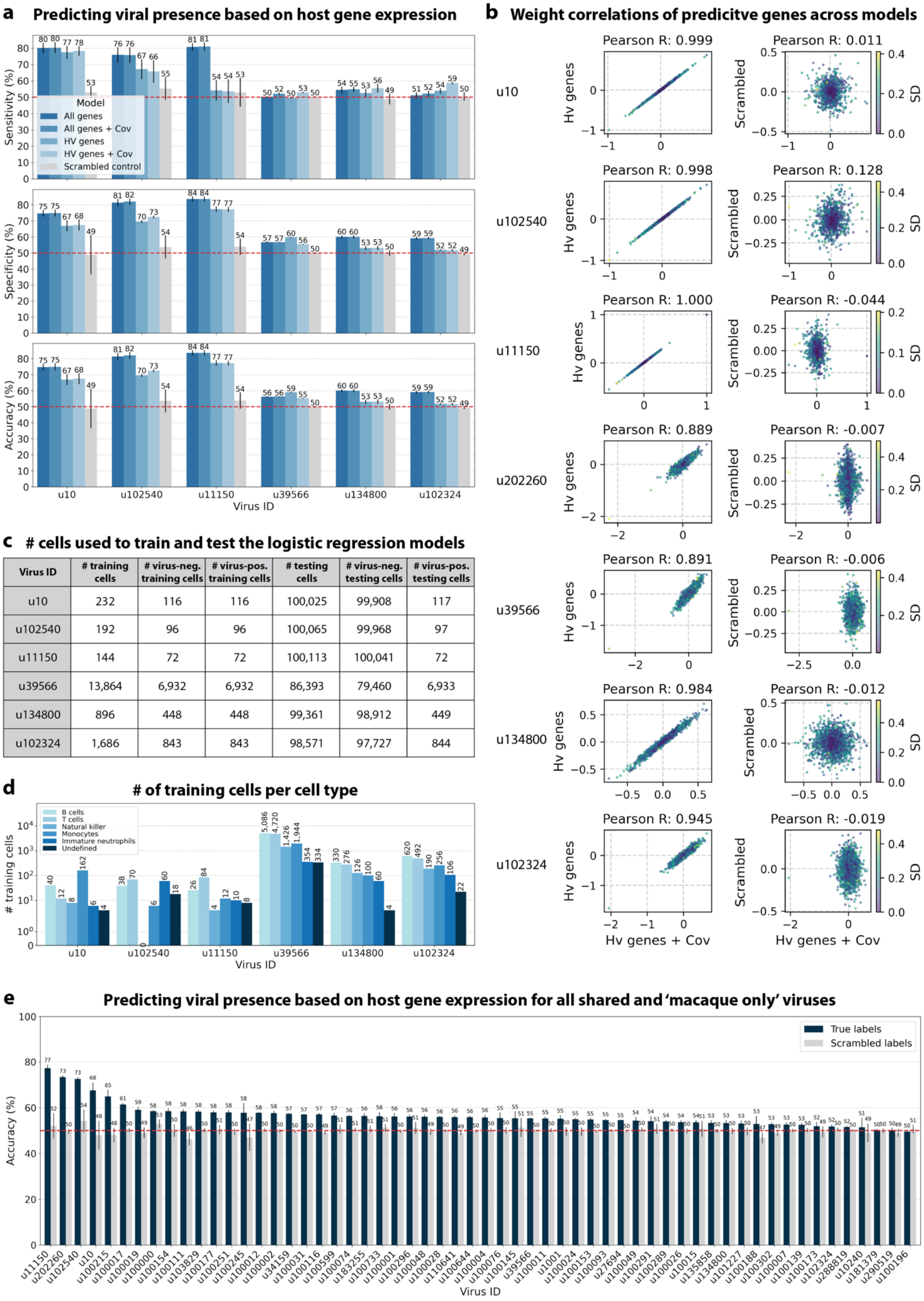
Extended Data Fig. 3.5: The number of positive cells for each individual virus ID obtained by different host masking options. Each virus ID shown here was observed in ≥ 0.05 % of cells. The host masking options are visualized in Extended Data Fig. 3.4a.



Extended Data Fig. 3.6: Number of positive cells per 10k cells for virus-like sequences from genera known to infect rhesus macaques[61] in the data from Kotliar et al.[37] analyzed using kallisto translated search with PalmDB. Host sequences were masked using the D-list option with the host genomes and transcriptomes, followed by host read capture using kallisto. No quality control thresholding of virus-like sequences was performed prior to generating this plot and the majority of these virus-like sequences were filtered out during quality control, and identification of contaminating sequences.



Extended Data Fig. 3.7: a, For each virus ID, the fraction of positive animal (top) and time point (bottom) samples was plotted. A sample was considered positive if at least 0.05 % of cells were positive. **b**, The number of positive cells for each virus ID or any combination of virus IDs for the count matrices generated from host-masked reads (D-list host genome and transcriptome + host transcriptome read capture) (left) and reads without any host masking (right). A large amount of reads for u202260 were masked when conservatively removing host reads (Fig. 3.5a). The plots were generated using PyVenn (https://github.com/tctianchi/pyvenn).



Virus ID	# training cells	# virus-neg. training cells	# virus-pos. training cells	# testing cells	# virus-neg. testing cells	# virus-pos. testing cells
u10	232	116	116	100,025	99,908	117
u102540	192	96	96	100,065	99,968	97
u11150	144	72	72	100,113	100,041	72
u39566	13,864	6,932	6,932	86,393	79,460	6,933
u134800	896	448	448	99,361	98,912	449
u102324	1,686	843	843	98,571	97,727	844

Figure legend on next page.

Extended Data Fig. 3.8: a, We trained logistic regression models to predict the presence of specific viruses based on host gene expression at single-cell resolution. The average accuracy, specificity, and sensitivity of the logistic regression models trained on highly variable (HV) or all macaque genes with or without donor animal and EVD time point as covariates are shown for ZEBOV (u10) and five novel virus-like sequences. Error bars indicate the standard deviation between models initialized with different random seeds. As a negative control, viral presence and absence labels were scrambled at random in the training data. **b**, Correlations of the average weights of predictive genes for models trained on HV genes with and without covariates on the real and scrambled labels. The weight correlations are lost when the model is trained using the scrambled labels. Virus IDs with high cell type specificity have slightly higher correlations than those with low cell type specificity. The color bar indicates the standard deviation (SD) of gene weights generated using different random seeds in the model trained on HV genes with covariates. The weights were max normalized between random seeds before computing the average and SD. **c**, The number of cells used to train and test the logistic regression models for ZEBOV (u10) and five novel virus-like sequences. **d**, Total number of training cells per cell type. The total consists of an equal number of virus-positive and -negative cells. **e**, Average prediction accuracy of models trained on HV genes with donor animal and EVD time point as covariates for all ‘macaque only’ and ‘shared’ viruses. Error bars indicate the standard deviation between models initialized with different random seeds.

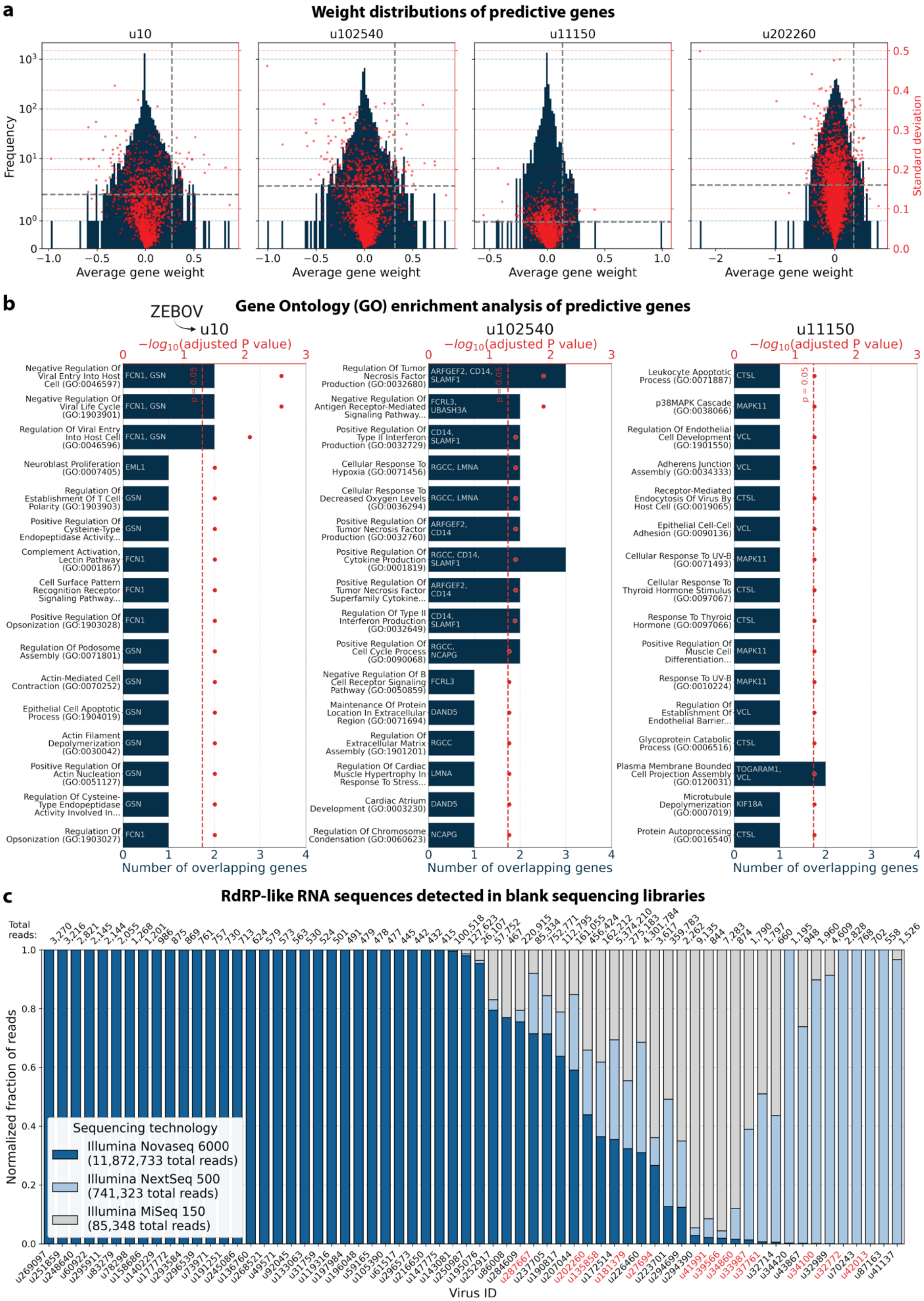


Figure legend on next page.

Extended Data Fig. 3.9: a, Average weight distributions of predictive genes from the models trained on highly variable genes with donor and time point as covariates for the four virus-like sequences with high predictive accuracy. The weights were averaged across models initialized using different random seeds and the standard deviations (SD) of the weights between seeds are shown in red. Gene weights were max normalized between random seeds before computing the average and SD. The dashed, grey lines indicate the minimum average gene weight and maximum SD for genes included in the enrichment analysis. **b**, Enrichment analysis of predictive genes from the regression model trained on highly variable genes with donor and time point as covariates. Approximately one third of the macaque Ensembl IDs did not have annotated gene names, which is a common problem for genomes from non-model organisms. We used gget[45] to translate annotated Ensembl IDs to gene symbols and to perform enrichment analysis using Enrichr[71] against the 2023 Gene Ontology (GO) Biological Processes database ('GO_Biological_Process_2023)[72]. Gene names are listed on the bar plot. Reported P values were corrected with the Benjamini-Hochberg method. **c**, Sequencing reads were obtained by sequencing multiple 'blank' sequencing libraries containing only sterile water and reagent mix. The plot shows the fraction of reads that map to different virus IDs for each sequencing technology. The fractions were normalized to the total number of reads obtained for each technology. The data was generated by Porter et al.[63] and analyzed using kallisto translated search with PalmDB. Virus IDs also detected in the macaque dataset are marked in red.

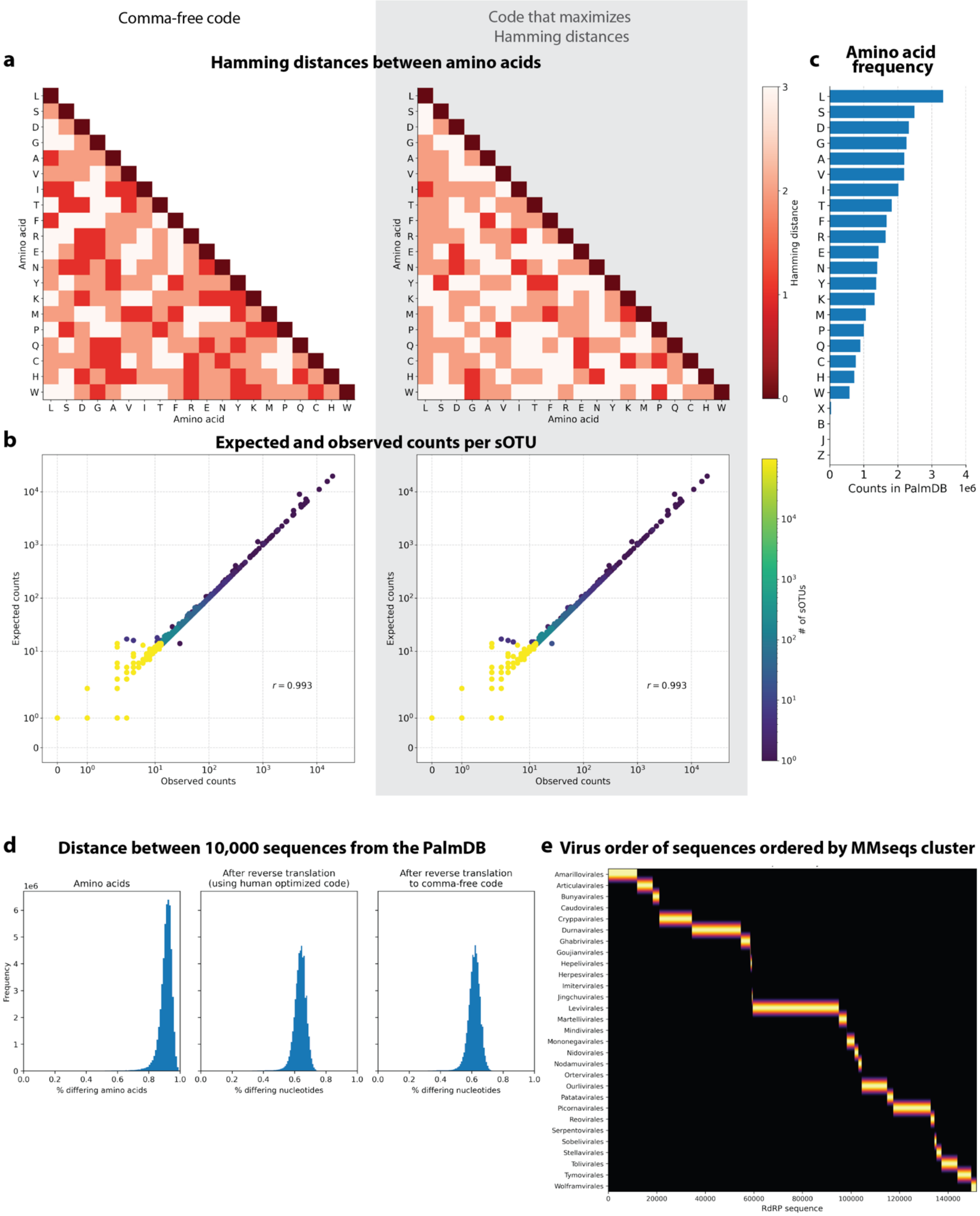


Figure legend on next page.

Extended Data Fig. 3.10: a, Hamming distances between amino acids in the comma-free code (left) and a second code that maximizes Hamming distances between amino acids that occur most often (right). **b**, We reverse translated all amino acid sequences in the PalmDB using the 'standard' genetic code (see Methods). The reverse translated PalmDB RdRP sequences were subsequently aligned to the optimized PalmDB amino acid reference (see Methods) with kallisto translated search. The left plot shows the expected and observed counts for each sOTU when kallisto performs the pseudoalignment in the comma-free code space. The plot on the right shows the expected and observed counts for each sOTU when kallisto performs the pseudoalignment using a second code that maximizes the Hamming distances between reverse translated amino acids. **c**, Occurrence of each amino acid in the PalmDB. **d**, Percentage of differing amino acids or nucleotides between 10,000 sequences randomly selected from the PalmDB before and after reverse translation using the standard genetic code (optimized for human) and comma-free code. **e**, The virus orders of RdRP sequences sorted based on their clustering by MMseqs2[79] (see Methods).

References

1. Mushegian, A. R. Are There 1031 Virus Particles on Earth, or More, or Fewer? *J. Bacteriol.* **202**, (2020).
2. Hendrix, R. W., Hatfull, G. F., Ford, M. E., Smith, M. C. M. & Burns, R. N. Evolutionary relationships among diverse bacteriophages and prophages: all the world's a phage. in *Horizontal gene transfer* 133–VI (Elsevier, 2002).
3. Anthony, S. J. *et al.* A strategy to estimate unknown viral diversity in mammals. *MBio* **4**, e00598–13 (2013).
4. Jones, K., Patel, N., Levy, M. *et al.* Global trends in emerging infectious diseases. *Nature* **451**, 990–993 (2008).
5. Mollentze, N., Babayan, S. A. & Streicker, D. G. Identifying and prioritizing potential human-infecting viruses from their genome sequences. *PLoS Biol.* **19**, e3001390 (2021).
6. Bjornevik, K. *et al.* Longitudinal analysis reveals high prevalence of Epstein-Barr virus associated with multiple sclerosis. *Science* **375**, 296–301 (2022).
7. Levine, K. S. *et al.* Virus exposure and neurodegenerative disease risk across national biobanks. *Neuron* **111**, 1086–1093.e2 (2023).
8. Cairns, D. M., Itzhaki, R. F. & Kaplan, D. L. Potential Involvement of Varicella Zoster Virus in Alzheimer's Disease via Reactivation of Quiescent Herpes Simplex Virus Type 1. *J. Alzheimers. Dis.* **88**, 1189–1200 (2022).
9. Edgar, R. C. *et al.* Petabase-scale sequence alignment catalyses viral discovery. *Nature* **602**, 142–147 (2022).
10. Babaian, A. & Edgar, R. Ribovirus classification by a polymerase barcode sequence. *PeerJ* **10**, e14055 (2022).
11. Chang, J.-T., Liu, L.-B., Wang, P.-G. & An, J. Single-cell RNA sequencing to understand host–virus interactions. *Virol. Sin.* (2023) doi:10.1016/j.virs.2023.11.009.
12. Hill, V. *et al.* Toward a global virus genomic surveillance network. *Cell Host Microbe* **31**, 861–873 (2023).
13. Melsted, P. *et al.* Modular, efficient and constant-memory single-cell RNA-seq preprocessing. *Nat. Biotechnol.* **39**, 813–818 (2021).
14. Ren, J. *et al.* Identifying viruses from metagenomic data using deep learning. *Quant Biol* **8**, 64–77 (2020).
15. Tithi, S. S., Aylward, F. O., Jensen, R. V. & Zhang, L. FastViromeExplorer: a pipeline for virus and phage identification and abundance profiling in metagenomics data. *PeerJ* **6**, e4227 (2018).
16. Camargo, A. P. *et al.* You can move, but you can't hide: identification of mobile genetic elements with geNomad. *bioRxiv* (2023) doi:10.1101/2023.03.05.531206.
17. Amgarten, D., Braga, L. P. P., da Silva, A. M. & Setubal, J. C. MARVEL, a Tool for Prediction of Bacteriophage Sequences in Metagenomic Bins. *Front. Genet.* **9**, 304 (2018).
18. Starikova, E. V. *et al.* Phigaro: high-throughput prophage sequence annotation. *Bioinformatics* **36**, 3882–3884 (2020).
19. Kieft, K., Zhou, Z. & Anantharaman, K. VIBRANT: automated recovery, annotation and curation of microbial viruses, and evaluation of viral community function from genomic sequences. *Microbiome* **8**, 90 (2020).

20. Antipov, D., Raiko, M., Lapidus, A. & Pevzner, P. A. Metaviral SPAdes: assembly of viruses from metagenomic data. *Bioinformatics* **36**, 4126–4129 (2020).
21. Ren, J., Ahlgren, N. A., Lu, Y. Y., Fuhrman, J. A. & Sun, F. VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome* **5**, 69 (2017).
22. Guo, J. *et al.* VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. *Microbiome* **9**, 37 (2021).
23. Xia, Y., Liu, Y., Deng, M. & Xi, R. Detecting virus integration sites based on multiple related sequencing data by VirTect. *BMC Med. Genomics* **12**, 19 (2019).
24. Bost, P. *et al.* Host-Viral Infection Maps Reveal Signatures of Severe COVID-19 Patients. *Cell* **181**, 1475–1488.e12 (2020).
25. Lee, C. Y. *et al.* Venus: An efficient virus infection detection and fusion site discovery method using single-cell and bulk RNA-seq data. *PLoS Comput. Biol.* **18**, e1010636 (2022).
26. Yasumizu, Y., Hara, A., Sakaguchi, S. & Ohkura, N. VIRTUS: a pipeline for comprehensive virus analysis from conventional RNA-seq data. *Bioinformatics* **37**, 1465–1467 (2021).
27. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biol.* **20**, 257 (2019).
28. Buchfink, B., Reuter, K. & Drost, H.-G. Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods* **18**, 366–368 (2021).
29. Golomb, S. W., Gordon, B. & Welch, L. R. Comma-Free Codes. *Canad. J. Math.* **10**, 202–209 (1958).
30. Crick, F. H., Griffith, J. S. & Orgel, L. E. CODES WITHOUT COMMAS. *Proc. Natl. Acad. Sci. U. S. A.* **43**, 416–421 (1957).
31. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161**, 1202–1214 (2015).
32. Ramsköld, D. *et al.* Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nat. Biotechnol.* **30**, 777–782 (2012).
33. Rosenberg, A. B. *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360**, 176–182 (2018).
34. Desai, N. *et al.* Temporal and spatial heterogeneity of host response to SARS-CoV-2 pulmonary infection. *Nat. Commun.* **11**, 6319 (2020).
35. Viloria Winnett, A. *et al.* Morning SARS-CoV-2 Testing Yields Better Detection of Infection Due to Higher Viral Loads in Saliva and Nasal Swabs upon Waking. *Microbiol Spectr* **10**, e0387322 (2022).
36. Viloria Winnett, A. *et al.* Extreme differences in SARS-CoV-2 viral loads among respiratory specimen types during presumed pre-infectious and infectious periods. *PNAS Nexus* **2**, gad033 (2023).
37. Kotliar, D. *et al.* Single-Cell Profiling of Ebola Virus Disease In Vivo Reveals Viral and Host Dynamics. *Cell* **183**, 1383–1401.e19 (2020).
38. Sharma, A. *et al.* Human iPSC-Derived Cardiomyocytes Are Susceptible to SARS-CoV-2 Infection. *Cell Rep Med* **1**, 100052 (2020).
39. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**, 357–359 (2012).

40. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *Gigascience* **10**, (2021).
41. Kuznetsov, A. & Bollin, C. J. NCBI Genome Workbench: Desktop Software for Comparative Genomics, Visualization, and GenBank Data Submission. *Methods Mol. Biol.* **2231**, 261–295 (2021).
42. Peck, K. M. & Lauring, A. S. Complexities of Viral Mutation Rates. *J. Virol.* **92**, (2018).
43. Sullivan, D. K. *et al.* kallisto, bustools, and kb-python for quantifying bulk, single-cell, and single-nucleus RNA-seq. *bioRxiv* 2023.11.21.568164 (2023) doi:10.1101/2023.11.21.568164.
44. Melsted, P. *et al.* Modular and efficient pre-processing of single-cell RNA-seq. *bioRxiv* 673285 (2019) doi:10.1101/673285.
45. Luebbert, L. & Pachter, L. Efficient querying of genomic reference databases with gget. *Bioinformatics* **39**, (2023).
46. Letunic, I. & Bork, P. Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. *Nucleic Acids Res.* **49**, W293–W296 (2021).
47. Lu, J. & Salzberg, S. L. Ultrafast and accurate 16S rRNA microbial community analysis using Kraken 2. *Microbiome* **8**, 124 (2020).
48. Kühl, M. A., Stich, B. & Ries, D. C. Mutation-Simulator: fine-grained simulation of random mutations in any genome. *Bioinformatics* **37**, 568–569 (2021).
49. Gihawi, A. *et al.* Major data analysis errors invalidate cancer microbiome findings. *MBio* e0160723 (2023).
50. Breitwieser, F. P., Pertea, M., Zimin, A. V. & Salzberg, S. L. Human contamination in bacterial genomes has created thousands of spurious proteins. *Genome Res.* **29**, 954–960 (2019).
51. Steinegger, M. & Salzberg, S. L. Terminating contamination: large-scale search identifies more than 2,000,000 contaminated entries in GenBank. *Genome Biol.* **21**, 115 (2020).
52. Wang, J. & Han, G.-Z. Genome mining shows that retroviruses are pervasively invading vertebrate genomes. *Nat. Commun.* **14**, 4968 (2023).
53. Hjörleifsson, K. E., Sullivan, D. K., Holley, G., Melsted, P. & Pachter, L. Accurate quantification of single-nucleus and single-cell RNA-seq transcripts. *bioRxiv* 2022.12.02.518832 (2022) doi:10.1101/2022.12.02.518832.
54. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv [q-bio.GN]* (2013).
55. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).
56. Camacho, C. *et al.* BLAST+: architecture and applications. *BMC Bioinformatics* **10**, 421 (2009).
57. Warren, W. C. *et al.* Sequence diversity analyses of an improved rhesus macaque genome enhance its biomedical utility. *Science* **370**, (2020).
58. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).
59. Svensson, V., da Veiga Beltrame, E. & Pachter, L. Quantifying the tradeoff between sequencing depth and cell number in single-cell RNA-seq. *bioRxiv* 762773 (2019)

doi:10.1101/762773.

60. Feldmann, H. & Geisbert, T. W. Ebola haemorrhagic fever. *Lancet* **377**, 849–862 (2011).
61. Wachtman, L. & Mansfield, K. Chapter 1 - Viral Diseases of Nonhuman Primates. in *Nonhuman Primates in Biomedical Research (Second Edition)* (eds. Abee, C. R., Mansfield, K., Tardif, S. & Morris, T.) 1–104 (Academic Press, 2012).
62. Ondov, B. D., Bergman, N. H. & Phillippy, A. M. Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics* **12**, 385 (2011).
63. Porter, A. F., Cobbin, J., Li, C.-X., Eden, J.-S. & Holmes, E. C. Metagenomic Identification of Viral Sequences in Laboratory Reagents. *Viruses* **13**, (2021).
64. Blomberg, J. *et al.* Phylogeny-directed search for murine leukemia virus-like retroviruses in vertebrate genomes and in patients suffering from myalgic encephalomyelitis/chronic fatigue syndrome and prostate cancer. *Adv. Virol.* **2011**, 341294 (2011).
65. Callanan, J. *et al. Rename one class (Leviviricetes - formerly Allassoviricetes), rename one order (Norzivirales - formerly Levivirales), create one new order (Timlovirales), and expand the class to a total of six families, 420 genera and 883 species.* http://rgdoi.net/10.13140/RG.2.2.25363.40481 (2021) doi:10.13140/RG.2.2.25363.40481.
66. Callanan, J. *et al.* Expansion of known ssRNA phage genomes: From tens to over a thousand. *Sci Adv* **6**, eaay5981 (2020).
67. Cohen, J. I. Herpesvirus latency. *J. Clin. Invest.* **130**, 3361–3369 (2020).
68. Woźniakowski, G. & Samorek-Salamonowicz, E. Animal herpesviruses and their zoonotic potential for cross-species infection. *Ann. Agric. Environ. Med.* **22**, 191–194 (2015).
69. Yao, X. *et al.* In Vitro Infection Dynamics of Wuxiang Virus in Different Cell Lines. *Viruses* **14**, (2022).
70. Valles, S. M. *et al.* ICTV Virus Taxonomy Profile: Iflaviridae. *J. Gen. Virol.* **98**, 527–528 (2017).
71. Kuleshov, M. V. *et al.* Enrichr: a comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res.* **44**, W90–7 (2016).
72. The Gene Ontology Consortium. The Gene Ontology knowledgebase in 2023. *Genetics*. **224**, 1 (2023).
73. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society*. **57**, 1 (1995).
74. Gierahn, T. M. *et al.* Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nat. Methods* **14**, 395–398 (2017).
75. Hughes, T. K. *et al.* Second-Strand Synthesis-Based Massively Parallel scRNA-Seq Reveals Cellular States and Molecular Features of Human Inflammatory Skin Pathologies. *Immunity* **53**, 878–894.e7 (2020).
76. Melsted, P., Ntranos, V. & Pachter, L. The barcode, UMI, set format and BUStools. *Bioinformatics* **35**, 4472–4473 (2019).
77. Pirtskhalava, M. *et al.* DBAASP v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic Acids*

Res. **49**, D288–D297 (2021).

78. Abdill, R. J. *et al.* Integration of 168,000 samples reveals global patterns of the human gut microbiome. *bioRxiv* (2023) doi:10.1101/2023.10.11.560955.
79. Hauser, M., Steinegger, M. & Söding, J. MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics* **32**, 1323–1330 (2016).
80. Lu, J. *et al.* Metagenome analysis using the Kraken software suite. *Nat. Protoc.* **17**, 2815–2839 (2022).
81. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).
82. Zulkower, V. & Rosser, S. DNA Chisel, a versatile sequence optimizer. *Bioinformatics* **36**, 4508–4509 (2020).
83. Gálvez-Merchán, Á. *et al.* Metadata retrieval from sequence databases with ffq. *Bioinformatics*, **39**, 1 (2023).
84. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
85. Ostendorf, B. N. *et al.* Common human genetic variants of APOE impact murine COVID-19 mortality. *Nature* **611**, 346–351 (2022).
86. Wick, R. R., Schultz, M. B., Zobel, J. & Holt, K. E. Bandage: interactive visualization of de novo genome assemblies. *Bioinformatics* **31**, 3350–3352 (2015).

Chapter 4

TRANSCRIPTOMICS IN NON-MODEL ORGANISMS – PART I

Challenges and Solutions

The first step in the analysis of single-cell RNA sequencing data following standard analysis workflows is the alignment of the data to a reference genome. This assumes that a reference genome for the species of interest is available. As discussed in Chapter 2, the number of viruses with the potential to cause human infectious disease is eclipsed by the comparatively few viruses with complete reference genomes. In the workflows described in Chapter 2, this challenge is overcome by identifying protein domains that were highly conserved across species. Assuming that a genome is available, the alignment quality will depend on the quality of the genome assembly, including the percentage of gaps in the assembly, annotation of protein-encoding and non-coding genes, including isoforms and gene candidates, and haplotype completeness. While there are high-quality genome assemblies for widely studied organisms such as human, mouse, and rhesus macaque, the quality of genome assemblies for other species quickly decreases (Figure 4.1).

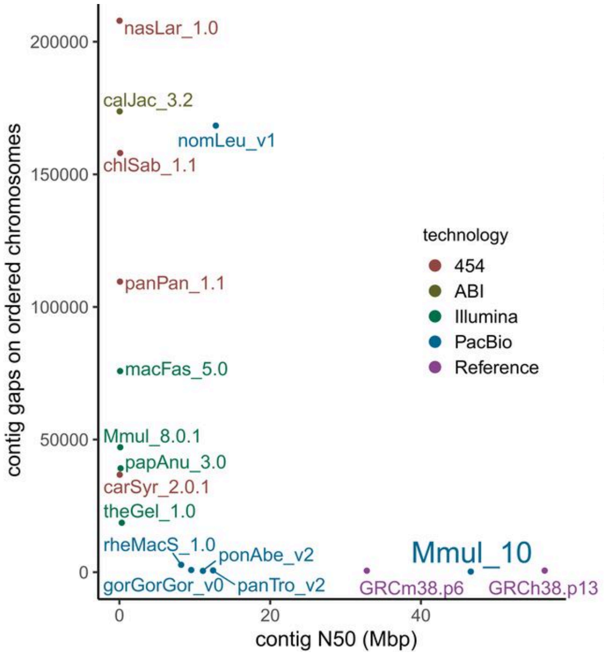
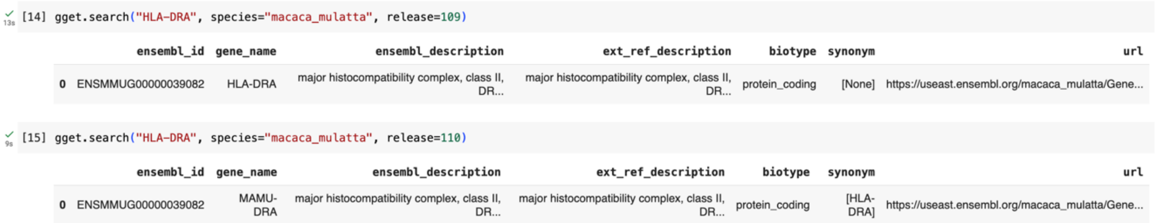


Figure 4.1 Number of gaps and contig N50 lengths for different mammalian genomes, including human (GRCh38.p13), mouse (GRCm38.p6), and rhesus macaque (Mmul_10). Reproduced from Warren *et al.*[1]

The following subchapter presents the results obtained through single-cell RNA sequencing of the non-model organism *Taeniopygia guttata* (zebra finch). This dataset was the first single-cell RNA sequencing dataset generated from zebra finch tissue. While the zebra finch was the second avian species, after chicken, to have its genome sequenced, at the time the analysis discussed below was performed, the zebra finch genome coverage was only 88.2x with a contig N50 of 12 Mb (compared to 67.8 Mb for the human genome assembly GRCh38). Moreover, depending on which assembly and version are used, different results may be obtained. This problem is further complicated when the difference between reference genomes is not documented comprehensively.



```
[14] gget.search("HLA-DRA", species="macaca_mulatta", release=109)
```

	ensembl_id	gene_name	ensembl_description	ext_ref_description	biotype	synonym	url
0	ENSMMUG00000039082	HLA-DRA	major histocompatibility complex, class II, DR...	major histocompatibility complex, class II, DR...	protein_coding	[None]	https://useast.ensembl.org/macaca_mulatta/Gene...

```
[15] gget.search("HLA-DRA", species="macaca_mulatta", release=110)
```

	ensembl_id	gene_name	ensembl_description	ext_ref_description	biotype	synonym	url
0	ENSMMUG00000039082	MAMU-DRA	major histocompatibility complex, class II, DR...	major histocompatibility complex, class II, DR...	protein_coding	[HLA-DRA]	https://useast.ensembl.org/macaca_mulatta/Gene...

Figure 4.2 Example commands and results obtained using the *gget search* module for the search term 'HLA-DRA,' showing how the annotation for this gene changed between two different Ensembl releases (109 and 110).

For example, between the zebra finch reference genomes GCA_003957565.2 (available on Ensembl) and GCA_003957565.4 (available on NCBI), the haplotypes were switched, with bTaeGut1_v1 containing the first haplotype (GCA_003957565.2) and bTaeGut1.4.pri the alternative haplotype (GCA_003957565.4). Both genomes should be derived from the male zebra finch 'Black17'. However, according to the fna files of bTaeGut1.4.pri (GCA_003957565.4) provided by RefSeq (GCF_003957565.2 as listed on https://www.ncbi.nlm.nih.gov/assembly/GCA_003957565.4), all chromosomes originated from the female zebra finch 'Blue55'. Moreover, only bTaeGut1.4.pri includes information from mitochondrial (MT) chromosomes, which is crucial for the assessment of cell health. The bTaeGut1.4.pri fna file provided by GenBank (GCA_003957565.4) is annotated as Black17, as expected (GCA_003957565.4_bTaeGut1.4.pri_genomic.fna). However, MT chromosomes are not immediately included, but available in a separate fna file. Additionally, chromosome W from the female bird Blue 55 is also not included in the general fna file, but is available when each chromosome is downloaded separately. This is not a problem for the analysis below, since all of those animals were male.

To make sense of the different assemblies and allow reproducible retrieval over time as assemblies get updated, I developed the *gget ref* module[3], which allows version-controlled retrieval of genome assemblies from Ensembl[2] (further described in Chapter 2).

Beyond incomplete genome sequence coverage, lower-quality reference genome assemblies tend to lack transcriptome annotations. In Ensembl genome assemblies, genes and transcripts are annotated with Ensembl IDs. For example, gene ENSG00000167360 encodes transcript ENST00000300778. Since these IDs do not contain any biological information, ideally, each ID has metadata associated with it, including the gene name and a description. However, in the zebra finch assembly GCA_003957565.2 (May 2019), 23.8 % of Ensembl IDs had no associated metadata. This often led to unannotated genes of interest obtained through clustering and differential gene expression analyses (described in the following subchapter). As a result, the Ensembl ID was the only information obtained about the gene of interest, which does not allow any further biological interpretation. This problem was the initial motivation behind writing the first *gget*[3] module, *gget info* (also see Chapter 2), which facilitates the retrieval of metadata about a gene from its Ensembl ID by combining information from several databases, including Ensembl[2], NCBI[4], and UniProt[5].

Combining information from different databases increases the chance of finding information for genes with little to no annotation on Ensembl and allows the comparison of information stored in each database. Moreover, *gget*'s database version arguments allow continued reproducibility over time as databases get updated (Figure 4.2 shows an example of a gene name change between Ensembl releases). The *gget search* module allows conversion in the other direction, from search terms or gene names to Ensembl IDs. Later modules, such as *gget blast* and *gget seq,* enable the retrieval of information about homologous genes in other species, which potentially have more extensively annotated reference genomes. Overall, the *gget* suite of tools allows leveraging reference genome annotation across databases and species, allowing the analysis and interpretation of sequencing data from species with sparsely annotated reference genomes.

Augmenting transcriptome metadata using *gget* does not solve the second problem of low-quality reference genomes: low coverage. Efforts such as the international Genome 10K (G10K) consortium[6] are working to develop cost-effective methods for producing high-quality, comprehensive reference genome assemblies. Moreover, the translated alignment algorithm described in Chapter 3 may be co-opted to align bulk and single-cell RNA sequencing data to reference proteomes from homologous species. Alignment in the amino acid space will make the alignment more robust to silent nucleotide substitutions between homologs and potentially allow the analysis of sequencing data from species with a missing or low-quality reference genome.

References

1. Warren, W. C. *et al.* Sequence diversity analyses of an improved rhesus macaque genome enhance its biomedical utility. *Science (80-.).* **370**, (2020).
2. Martin, F. J. *et al.* Ensembl 2023. *Nucleic Acids Res.* **51**, D933–D941 (2023).
3. Luebbert, L. & Pachter, L. Efficient querying of genomic reference databases with gget. *Bioinformatics* **39**, 4–6 (2023).
4. Sayers, E. W. *et al.* Database resources of the national center for biotechnology information. *Nucleic Acids Res.* **50**, D20–D26 (2022).
5. Consortium, T. U. UniProt: the Universal Protein Knowledgebase in 2023 - Google Scholar. **51**, 523–531 (2023).
6. Rhie, A. *et al.* Towards complete and error-free genome assemblies of all vertebrate species. *Nature* **592**, 737–746 (2021).

Neuronal Dynamics of Behavior Recovery in Zebra Finches

Preamble

Here, single-cell RNA sequencing is used to investigate the neuronal dynamics underlying the recovery of a learned behavior after chronically silencing inhibitory neurons in zebra finches. Beyond the challenges of studying a non-model organism described in the previous subchapter, this analysis was further complicated by the experimental design comparing two conditions. Here, I am only including the relevant single-cell RNA sequencing results and methods from the published paper.

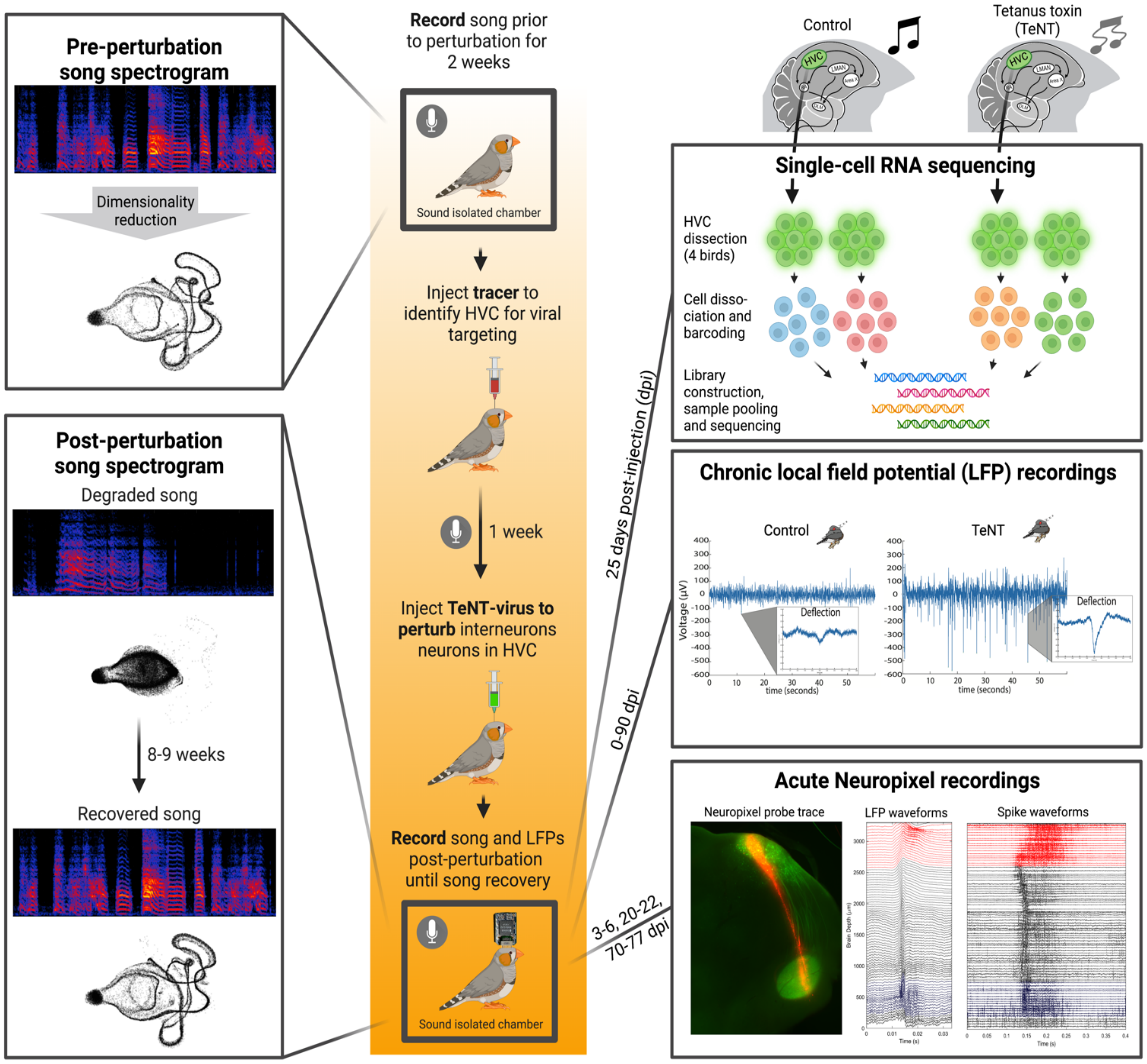
Zsofia Torok, **Laura Luebbert**, Jordan Feldman, Alison Duffy, Alexander A. Nevue, Shelyn Wongso, Claudio V. Mello, Adrienne Fairhall, Lior Pachter, Walter G. Gonzalez, Carlos Lois (2023). Recovery of a learned behavior despite partial restoration of neuronal dynamics after chronic inactivation of inhibitory neurons. *bioRxiv*. https://doi.org/10.1101/2023.05.17.541057

Summary

Maintaining motor skills is crucial for an animal's survival, enabling it to endure diverse perturbations throughout its lifespan, such as trauma, disease, and aging. What mechanisms orchestrate brain circuit reorganization and recovery to preserve the stability of behavior despite the continued presence of a disturbance? To investigate this question, we chronically silenced inhibitory neurons, which altered brain activity and severely perturbed a complex learned behavior for around two months, after which it was precisely restored. Electrophysiology recordings revealed abnormal offline dynamics resulting from chronic inhibition loss, while subsequent recovery of the behavior occurred despite partial normalization of brain activity. Single-cell RNA sequencing revealed that chronic silencing of interneurons leads to elevated levels of microglia and MHC I. These experiments demonstrate that the adult brain can overcome extended periods of drastic abnormal activity. The reactivation of mechanisms employed during learning, including offline neuronal dynamics and upregulation of MHC I and microglia, could facilitate the recovery process following perturbation of the adult brain. These findings indicate that some forms of brain plasticity may persist in a dormant state in the adult brain until they are recruited for circuit restoration.

Introduction

Maintaining the ability to precisely execute motor behavior throughout life, despite perturbations due to trauma, disease, or aging, is crucial for reproduction and survival. To reliably execute behaviors, brain circuits require a balance of excitation and inhibition (E/I balance) to maintain physiological activity patterns. Loss of E/I balance causes abnormal patterns of neuronal activity, which can result in diseases such as epilepsy [1,2]. Given its importance, brain circuits strive to restore E/I balance once disturbed [3]. However, the


Pre-perturbation song spectrogram
Dimensionality reduction
Post-perturbation song spectrogram
Degraded song
8-9 weeks
Recovered song
Record song prior to perturbation for 2 weeks
Sound isolated chamber
Inject tracer to identify HVC for viral targeting
1 week
Inject TeNT-virus to perturb interneurons neurons in HVC
Record song and LFPs post-perturbation until song recovery
Sound isolated chamber
25 days post-injection (dpi)
0-90 dpi
3-6, 20-22, 70-77 dpi
Control
Tetanus toxin (TeNT)
HVC
LMAN
Area X
Single-cell RNA sequencing
HVC dissection (4 birds)
Cell disso-ciation and barcoding
Library construction, sample pooling and sequencing
Chronic local field potential (LFP) recordings
Control
TeNT
Voltage (µV)
Deflection
time (seconds)
Acute Neuropixel recordings
Neuropixel probe trace
LFP waveforms
Spike waveforms
Brain Depth (µm)
Time (s)

Graphical Abstract Schematic overview of the experiments performed in this study. To investigate how a complex motor behavior recovers after chronic loss of inhibitory tone, we blocked the function of zebra finch HVC inhibitory neurons by bilateral stereotaxic injection of an AAV viral vector into HVC. Throughout various timepoints in this perturbation paradigm, we recorded song behavioral data, electrophysiological measurements (chronic and acute within HVC), and measured changes in gene expression at single-cell resolution.

mechanisms orchestrating circuit reorganization and recovery of E/I balance during the continued presence of a perturbation remain poorly understood.

Zebra finches produce a highly stereotyped song with minimal variability over extended periods of time [4], underpinned by temporally precise neural activity [5]. This species thus provides an optimal model to simultaneously track abnormal brain activity, its effect on behavior, and changes to both over time. It serves as an excellent model for studying chronic E/I imbalance and accompanying changes over time at the behavioral, neuronal, and transcriptomic levels.

Here, we genetically block inhibitory neurons in HVC (proper name), a premotor brain nucleus of the male zebra finch involved in song production, to chronically perturb the E/I balance. HVC contains two main types of excitatory neurons that project to two main downstream targets: nucleus X (proper name) and nucleus RA (robust nucleus of the arcopallium). In addition, HVC includes several types of inhibitory neurons whose axons do not leave HVC; thus, they act locally within HVC [5]. Juvenile male zebra finches learn their song from their fathers during the "critical period" and once learned, produce a highly stereotypical song for the rest of their lives [6]. Previous work has shown that inhibition plays a role during development to close the critical period and protect learned components of the song in juvenile animals [7]. Acutely blocking interneuron signaling in adult animals by chemicals leads to abnormal behaviors that quickly return to normal once the chemical is washed out [8]. However, it is not known how long-term disruption of inhibitory neurons affects neuronal dynamics, and whether behaviors can recover after such drastic perturbation.

Results

To investigate how a complex motor behavior recovers after chronic loss of inhibitory tone, we blocked the function of HVC inhibitory neurons in adult male zebra finches by bilateral stereotaxic injection of an AAV viral vector into HVC. The AAV viral vector carried the light chain of tetanus toxin (TeNT), driven by the human dlx5 promoter, which is selectively active in inhibitory neurons [9]. TeNT blocks the release of neurotransmitters from presynaptic terminals, thereby preventing neurons from communicating with their postsynaptic partners [10]. Thus, expression of TeNT does not directly alter the ability of neurons to fire action potentials, but effectively mutes them. As a control, a second group of animals was injected with an AAV carrying the green fluorescent protein NeonGreen driven by the ubiquitous promoter CAG. Throughout various time points in this perturbation paradigm, we recorded song behavioral data, obtained electrophysiological measurements (chronic and acute within HVC), and measured changes in gene expression at single-cell resolution (Graphical Abstract).

Single-cell RNA sequencing suggests mechanisms of neuronal plasticity driven by microglia and MHC class I genes during song perturbation

To investigate cellular mechanisms that might underlie the observed changes in neuronal activity and behavior at the transcriptomic level, we performed single-cell RNA sequencing (scRNAseq) of HVC from control (n=2) and TeNT-treated (n=2) adult male zebra finches at 25 dpi, around the time of peak song distortion. HVC from both hemispheres of all four birds were dissected based on retrograde tracer fluorescence and dissociated to prepare single-cell suspensions, which were indexed and pooled. This allowed the construction of a combined dataset, containing results from all organisms and conditions, without the need for batch correction (Supplementary Figure 4.1, Supplementary Table 4.1). Following quality control, we retained a total of 35,804 single-cell profiles spanning four individuals, consisting of two control and two TeNT-treated animals.

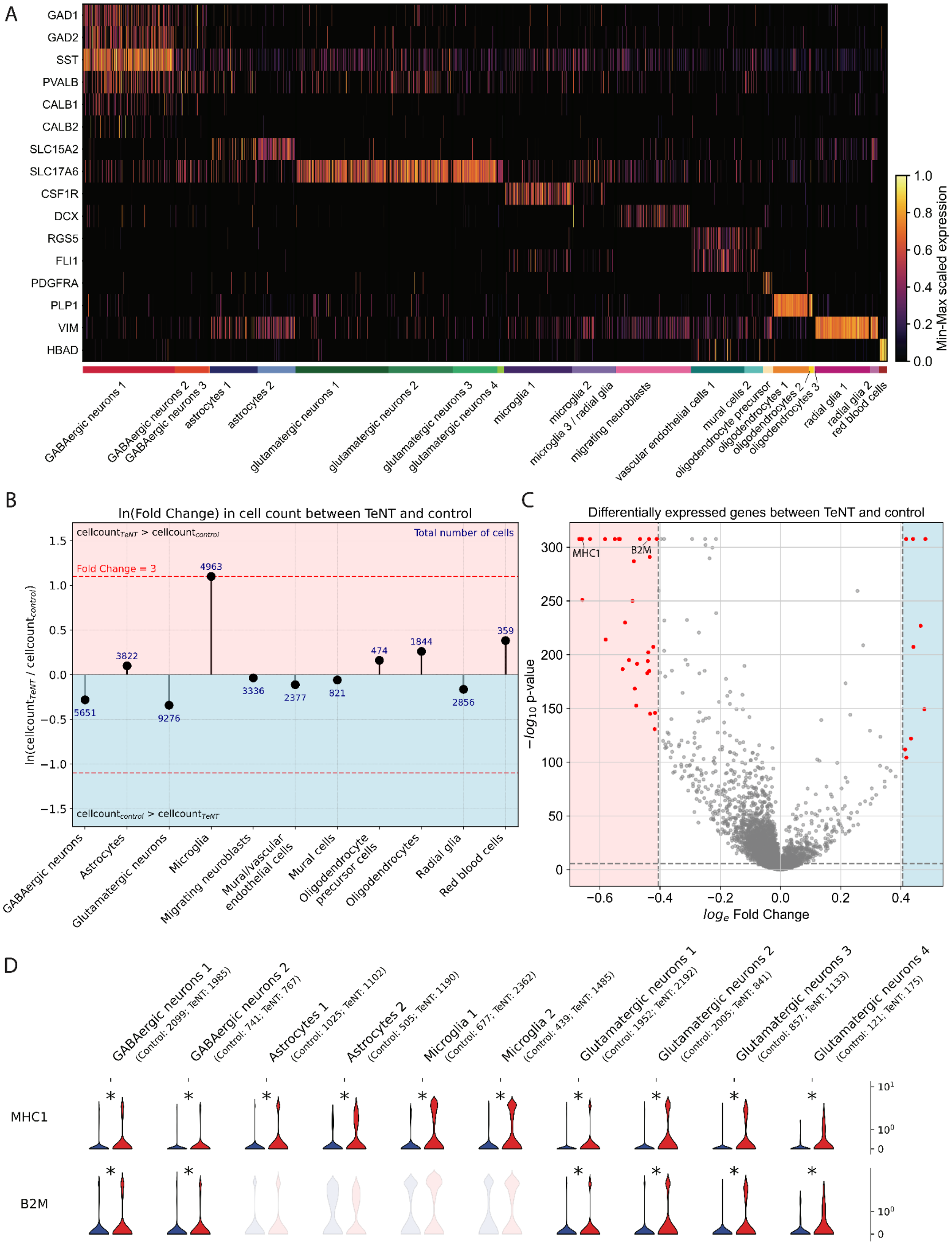


Figure legend on next page.

Figure 4.3 Transcriptomic changes at single-cell resolution in HVC at 25 days after chronic loss of inhibitory neurons through viral expression of tetanus toxin (TeNT). Single-cell RNA sequencing of the HVCs of control (n=2) and TeNT-treated (n=2) animals was performed at 25 days post-injection (dpi). **A** Heatmap showing min-max scaled expression of cell type marker genes for each cell type (data from both control and TeNT-treated animals). **B** Log-fold change in total number of cells per cell type between TeNT-treated and control animals. **C** Volcano plot showing statistical significance over magnitude of change of differentially expressed genes between TeNT-treated and control animals across all cell types. Dotted lines indicate fold change = 1.5 and p value = Bonferroni corrected alpha of 0.05. A list of all differentially expressed genes can be found here: https://github.com/lauraluebbert/TL_2023. **D** Violin plots of normalized counts of major histocompatibility complex 1 α chain-like (MHC1) (ENSTGUG00000017273.2) and beta 2 microglobulin-like (B2M) (ENSTGUG00000004607.2) genes in control (n=2, blue) and TeNT-treated (n=2, red) animals per cell cluster. A star indicates a significant increase in gene expression in TeNT-treated animals compared to control ($p < 0.05$ and fold change > 1.5).

While cell type abundance was highly concordant between replicates of the same condition (Supplementary Figure 4.1, Figure 4.3 A), we found that animals treated with TeNT we found that animals treated with TeNT displayed a three-fold increase in the number of microglia (Figure 4.3 B). This increase in microglia was likely not due to an inflammatory reaction caused by the surgical procedure or the AAV injection because control animals also received a viral injection with a highly similar construct. Thus, we hypothesize that the increase in microglia in TeNT-treated animals is a consequence of the chronic muting of inhibitory neurons.

Several studies have shown that microglia play a role in synaptic plasticity during early brain development and learning in mammals [32–35]. These prior observations in combination with our findings suggest that microglia might participate in the synaptic reorganization triggered by circuit perturbation. We performed in situ hybridization (ISH) using a probe against RGS10, a gene expressed in microglia, during song degradation at 25 dpi and after recovery at 90 dpi. At 25 dpi, the number of microglia increased in TeNT-treated animals compared to control (Figure 4.4 A, Supplementary Figure 4.2 A and B), and returned to control levels by 90 dpi, when the song had recovered.

To further investigate the hypothesis that microglia are associated with circuit reorganization involved in neuronal plasticity, we counted the number of microglial cells in HVC at different times during song learning in naive (untreated), juvenile birds using ISH. The number of microglia in the HVC of juveniles was higher during the early stages of the song learning period (30-50 days post-hatching (dph)), compared to 70 dph, after the song became more stereotypic (Figure 4.4 B, Supplementary Figure 4.2 C and D).

Furthermore, scRNAseq analysis revealed a significant increase in the expression of the α chain of major histocompatibility complex class I (MHC I) and β2-microglobulin (B2M) across several neuronal cell types in TeNT-treated animals (Figure 4.3 C and D) and confirmed the increases in MHC I by ISH (Figure 4.4 C, Supplementary Figure 4.2 E and F). MHC class I molecules are heterodimers that consist of two polypeptide chains, α and

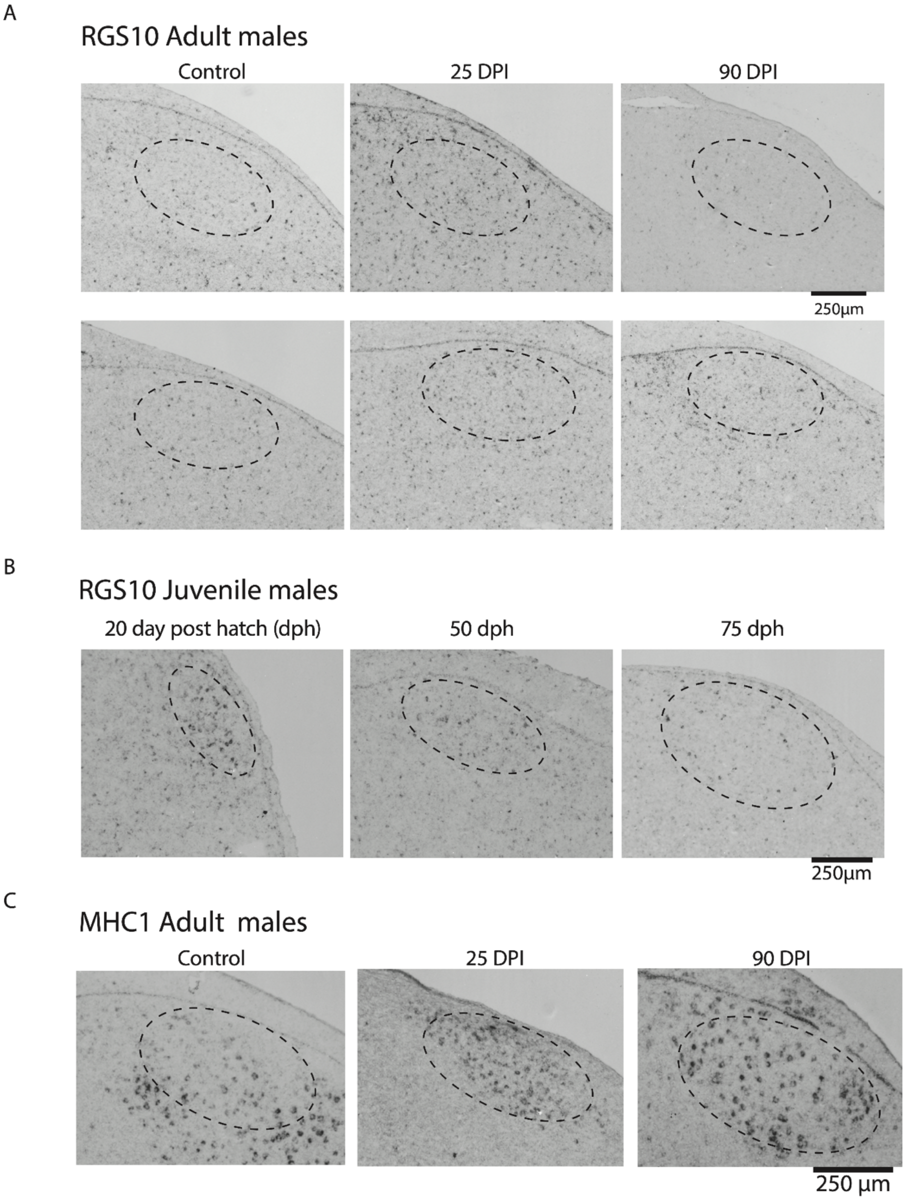


Figure 4.4 *In situ* hybridization of microglia marker gene RGS10 in adult male control, TeNT-treated and juvenile male HVC & MHC1 gene in adult male control and TeNT-treated HVC. **A** Histological sections of HVC (in control and TeNT-treated animals at 25 and 90 dpi) after *in situ* hybridization of RNA probes for RGS10 (a gene marker for microglia). **B** Histological sections of HVC in naive juvenile males (at 20, 50, and 75 days post-hatching (dph)) after *in situ* hybridization of RNA probes for RGS10. **C** Histological sections of HVC (from control and TeNT-treated animals at 25 and 90 dpi) after *in situ* hybridization of RNA probes for MHC1. Black/darker dots indicate enzyme reactions resulting in successful probe localization and suggest target gene expression.

B2M, and are involved in antigen presentation for T cells [36]. This increase in MHC I and B2M is likely triggered by the genetic silencing of inhibitory neurons, and not due to an inflammatory response because it was not observed in control animals injected with a control virus. MHC I and B2M have been observed in previous studies to be highly expressed in neurons during brain development [37,38], consistent with a hypothetical role in synaptic plasticity [39,40]. Here we observe that MHC I and B2M are upregulated in response to perturbation of neuronal activity in a fully formed circuit, indicating their possible role in the restoration of brain function.

Discussion

The brain requires a balance between excitation and inhibition to maintain physiological activity patterns and enable reliable behaviors. We found that chronic muting of inhibitory neurons in a pre-motor circuit severely disrupts a learned motor behavior for an extended period of time (30-90 days). Even after several weeks of perturbed brain activity, the brain circuit is able to regain function and recover the behavior, despite failing to restore some aspects of its neuronal dynamics. We propose that the return to control-like offline voltage deflections and the precision of local neuronal activity during alpha oscillations during these deflections are key components that accompany the behavioral recovery. Our observations indicate a putative relationship between activity dynamics offline and the restoration of circuit function. Furthermore, our data suggest that microglia and MHC I may be involved in changes caused by chronic perturbation of neuronal activity. These experiments reveal that the adult brain can overcome extended periods of E/I imbalance, potentially by processes that occur offline. The reactivation of mechanisms typically employed during juvenile learning, including night replay and activation of MHC I and microglia, could facilitate the recovery process following perturbation in the adult brain. This indicates that some forms of circuit plasticity may persist throughout adulthood, entering a dormant state until their activation is required for circuit restoration.

Methods

Animals

All procedures involving zebra finches were approved by the Institutional Animal Care and Use Committee of the California Institute of Technology. All birds used in the current study were bred in our own colony and housed with multiple conspecific cage mates of mixed genders and ages until used for experiments. Before any experiments, adult male birds (>120 days post-hatching (dph)) were singly housed in sound isolation cages with a 14/10 hr light/dark cycle for >5 days until they habituated to the new environment and started singing. Thereafter, birds were kept in isolation until the end of the experiment.

Behavioral recordings

Adult male zebra finches' (n=30, 130-890 dph) undirected songs were recorded 24/7 in sound-isolated chambers for 10-14 days before any manipulation to get a baseline of their song. Recordings were done with microphones (Audio-technica, AT831b) that are connected to an amplifier M-TRACK 8 and recording software Sound Analysis Pro 2011 at 44100 Hz. Animals were housed in these chambers and continuously recorded for the duration of the experiments.

Viral vectors

AAV-TeNT contained the promoter from the human dlx5 gene driving expression of the light chain of tetanus toxin fused to EGFP with a PEST domain. AAV9-dlx-TeNT was obtained from the Duke viral core facility. The control virus used was AAV9-CAG-NeonGreen, where CAG drives the expression of NeonGreen which is a GFP variant.

Stereotaxic injection

Birds were anesthetized with isoflurane (0.5% for initial induction, 0.2% for maintenance) and head-fixed on a stereotaxic apparatus. First, to inject a retrograde tracer in area X, craniotomies were made bilaterally and fluorescent tracers (fluoro-ruby 10%, 100-300 nL) were injected through a glass capillary (tip size ~25 µm) into the corresponding nuclei (coordinates from dorsal sinus in mm - area X: Anteroposterior (AP) 3.3-4.2, Mediolateral (ML) 1.5-1.6, Deep (D): 3.5-3.8). To deliver the virus (AAV) into HVC, a second surgery was performed 7-10 days after retrograde tracer injection. By then, HVC was strongly labeled by fluorescence and visible through a fluorescent stereoscope. AAVs diffuse extensively (~500 µm), and a single injection (~100 nL) in the center of HVC was sufficient to label enough cells. All injections in HVC were performed at ~20 nL/min to minimize physical damage. At the end of every surgery, craniotomies were covered with Kwik-Sil, and the skin incision was closed with Gluture.

Chronic electrophysiology recordings

Animals (n=4, 300-700 dph) were implanted in the right hemisphere HVC with 4 by 4 electrode arrays (Neuronexus A4x4-3mm-50/100-125-703-CM16LP) based on retrograde fluorescent labeling of HVC (just as for viral injections). Post-perfusion histology images were obtained to locate the electrode array within HVC for each animal (Supplementary Figure 4.3). Electrode implantation occurred within the same surgery as the viral injection.

This procedure follows the same surgical steps as the viral delivery protocol, until the point of electrode implantation. A small opening was cut on the dura (just big enough to fit the electrode array) to lower the electrodes manually. The reference and ground were a gold screw pin placed into the cerebellum. The skin was removed from the surface of the skull for the majority of the surface, in order to secure the implant. Before implantation, the skull and the craniotomies were cleaned with saline and dried and the skull was prepared according to the protocol of the C&B Metabond cement system. Post implantation we covered the craniotomies with kwik-sil. Once hardened, we covered the whole skull, and the part of the electrode still exposed, with metabond. The head stage (Intan RHD Part # C3335) was connected to the probe before implantation and securely metabonded to the connection between the probe and head stage in order to prevent detachment when the bird is moving. SPI interface cables (Intan Part #C3203, #C3213) were connected to the acquisition board (Open Ephys). Data was recorded at 30,000 Hz with the Open Ephys software system. Animals were freely moving with a passive counterweight-based commutator system.

Acute electrophysical recordings

Animals (n=10, 140-250 dph) went through the same surgical procedure as described for a stereotaxic viral injection. However, at the end of the surgery the skin was removed from the skull, and the whole skull was pre-treated and covered in metabond except for the craniotomies over HVC that were covered with kwik-cast until the day of the acute recording session. Shortly before the recording session, a head-bar was glued on top of the frontal surface of the metabonded skull to allow the head-fixation of the bird for the recording session. Then, the kwik-cast was removed from the craniotomy over HVC (left or right hemisphere or both depending on the animal) and a small incision was made in the dure over HVC, which was identified by the retrograde tracer previously injected. The ground was placed into the cerebellum. Then the high-density silicone probe (Neuropixel) was lowered with a motorized arm over hours for 2.6-3 mm deep into the brain. The head stage and acquisition board was connected to the computer and data was recorded with the Open Ephys software. Once the probe settled in the brain, we had 4 distinct recording sessions. Post-perfusion histology images were obtained to locate electrode array within HVC for each animal (Supplementary Figure 4.4). Recording sessions: lights on silence (10 min), followed by playback of the bird's own song (3-10 min); lights-off silence (10 min), followed by playback of the bird's own song (3-10 min); microinjection of 100 nL 250 μM Gabazine (Hellobio, HB0901), followed by the same protocol of lights-off and on without Gabazine.

Single-cell RNA sequencing
Animals
All of the work described in this study was approved by California Institute of Technology and Oregon Health & Science University's Institutional Animal Care and Use Committee and is in accordance with NIH guidelines. Zebra finches (*Taeniopygia guttata*) were obtained from our own breeding colony or purchased from local breeders.

Dissociation and cDNA generation
Animals were anesthetized with a mix of ketamine-xylazine (0.02 mL / 1 gram) and quickly decapitated, then the brain was placed into a carbogenated (95% O2, 5% CO2) NMDG-ACSF petri dish on ice. The brains were dissected on a petri dish with NMDG-ACSF surrounded by ice under an epifluorescent microscope guided by the fluoro-ruby retrograde tracing from Area X to HVC.

We used the commercially available Worthington Papain Dissociation system with some minor changes and add-on steps. We followed all the steps included in the Worthington protocol with a final concentration of 50 U/mL of papain. To match the intrinsic osmolarity of neurons in zebra finches we used NMDG-ACSF (~310 mOsm) instead of the EBSS for post-dissection and STOP solution. Another modification was to add 20 μL of 1 mg/mL Actinomycin D (personal communication from Allan-Hermann Pool; 47) into 1 mL of the post-dissection medium and the STOP solution in which trituration occurred. Papain digestion occurred for an hour on a rocking surface with constant carbogenation in a secondary container above the sample vial at RT. We performed trituration with increasingly smaller diameter glass pasteur pipettes. Trituation was performed inside the papain solution. Then, once the tissue was fully dissociated, we centrifuged the samples at 300 g RT for 5 minutes and resuspended them in STOP solution. Next, we used a 40 μm Falcon cell strainer pre-wet with the STOP solution and centrifuged again at 300 g RT for 5 min. Finally, we resuspended the cell pellet in 60μl of STOP solution and proceeded to barcoding and cDNA synthesis. The cell barcoding, cDNA synthesis, and library generation protocol were performed according to the Chromium v3.1 next GEM single cell 3’ reagent kits by Jeff Park in the Caltech sequencing facility. Sequencing was performed on an Illumina Novaseq S4 sequencer with 2x150 bp reads.

Generation of count matrices

The reference genome GCA_003957565.2 (Black17, no W) was retrieved from Ensembl on March 20, 2021 (http://ftp.ensembl.org/pub/release-104/gtf/taeniopygia_guttata/). We quantified the gene expression in each of the four datasets using the kallisto-bustools workflow [48]. The reference index was built using the kb-python (v0.26.3) ref command and the above-mentioned reference genome. Subsequently, the WRE sequence was manually added to the cdna and t2g files generated by kallisto-bustools to allow the identification of transgenic cells. The count matrix was generated for each dataset using the kallisto-bustools count function. The resulting count matrices were compared to those generated by the 10X Cell Ranger pipeline (v6.0.1) and kallisto-bustools count with multimapping function. For all four datasets, kallisto-bustools mapped approximately 10% more reads than Cell Ranger (Supplementary Figure 4.5). No increase in confidently mapped reads was observed when using the multimapping function, indicating that reads align confidently to one gene in the reference genome (Supplementary Figure 4.5).

Quality control and filtering

The datasets were filtered separately based on the expected number of cells and their corresponding minimum number of UMI counts (Supplementary Figure 4.6). Following quality control based on apoptosis markers and library saturation plots (Supplementary Figure 4.6), the count matrices were concatenated and normalized using log(CP10k + 1) for downstream dimensionality reduction and visualization using Scanpy’s (v1.9.1) [49] normalize_total with target sum 10,000 and log1p. Gene names and descriptions for Ensembl IDs without annotations were obtained using gget (v0.27.3) [50].

Dimensionality reduction and normalization

The concatenated data was mapped to a lower dimensional space by PCA applied to the log-normalized counts filtered for highly variable genes using Scanpy’s highly_variable_genes. Next, we computed nearest neighbors and conducted Leiden clustering [51] using Scanpy.

Initially, this approach was performed on the control and TeNT datasets separately. This resulted in the identification of 19 clusters in the control data and 22 clusters in the TeNT data (Supplementary Figure 4.6). For both conditions, equal contribution from both datasets indicated that there was minimal batch effect, as expected since the data was sequenced in a pooled sequencing run. We also performed batch correction using scVI [52] which did not change the contribution of each dataset per cluster. As a result, we continued the analysis using the data that was not batch-corrected with scVI.

Next, we concatenated all four datasets and followed the approach described above. This resulted in the identification of 21 Leiden clusters, which we also refer to as cell types (Figure 4.3 A). Each cluster was manually annotated with a cell type based on the expression of previously established marker genes [53]. The cell type annotation was validated by the top 20 differentially expressed genes extracted from each cluster using Scanpy's rank_genes_groups (P values were computed using a t-test and adjusted with the Bonferroni method for multiple testing) (Supplementary Figure 4.1). Clusters identified as glutamatergic neurons were further broken down into HVC-X- and HVC-RA-projecting glutamatergic neurons using previously established marker genes (data not shown; also see https://github.com/lauraluebbert/TL_2023). We found that reclustering all cells labeled as glutamatergic neurons using the Leiden algorithm did not yield different results and we therefore continued with the initial clusters (data not shown). All results discussed in this paper were confirmed by both jointly and separately clustering the experimental conditions.

Comparative analysis of clusters and conditions

Differentially expressed genes between clusters were identified using Scanpy's rank_genes_groups (p values were computed using a t-test and adjusted with the Bonferroni method for multiple testing, and confirmed by comparison to P values generated with Wilcoxon test with Bonferroni correction).

In the violin plots, unless otherwise indicated, a star indicates a p value < 0.05 and a fold change > 1.5 difference in mean gene expression between the indicated conditions (p value computed with scipy.stats' (v1.7.0) ttest_ind and adjusted with the Bonferroni method for multiple testing).

In situ hybridization

Animals

All of the work described in this study was approved by the California Institute of Technology and Oregon Health & Science University's Institutional Animal Care and Use Committee and is in accordance with NIH guidelines. Zebra finches (*Taeniopygia guttata*) were obtained from our own breeding colony or purchased from local breeders. Developmental gene expression in HVC in the 20-, 50-, and 75-days post-hatch (dph) male and female zebra finches was assessed as previously described [54]. The sex of birds was determined by plumage and gonadal inspection. Birds were sacrificed by decapitation, bisected in the sagittal plane and flash-frozen in Tissue-Tek OCT (Sakura-Finetek), and frozen in a dry ice/isopropyl alcohol slurry. Brains of TeNT-manipulated finches were coronally blocked anterior to the tectum and flash frozen in Tissue-Tek (Sakura). All brains were sectioned at 10 µm on a cryostat and mounted onto charged slides (Superfrost Plus, Fisher).

In situ hybridization

In situ hybridization was performed as previously described [55,56]. Briefly, DIG-labeled riboprobes were synthesized from cDNA clones for RGS10 (CK312091) and LOC100231469 (class I histocompatibility antigen, F10 alpha chain; DV951963). Slides containing the core of HVC were hybridized overnight at 65°C. Following high stringency washes, sections were blocked for 30 min and then incubated in an alkaline phosphatase conjugated anti-DIG antibody (1:600, Roche). Slides were then washed and developed overnight in BCIP/NBT chromogen (Perkin Elmer). To minimize experimental confounds between animals, sections for each gene were fixed together in 3% paraformaldehyde, hybridized with the same batch of probe, and incubated in chromogen for the same duration.

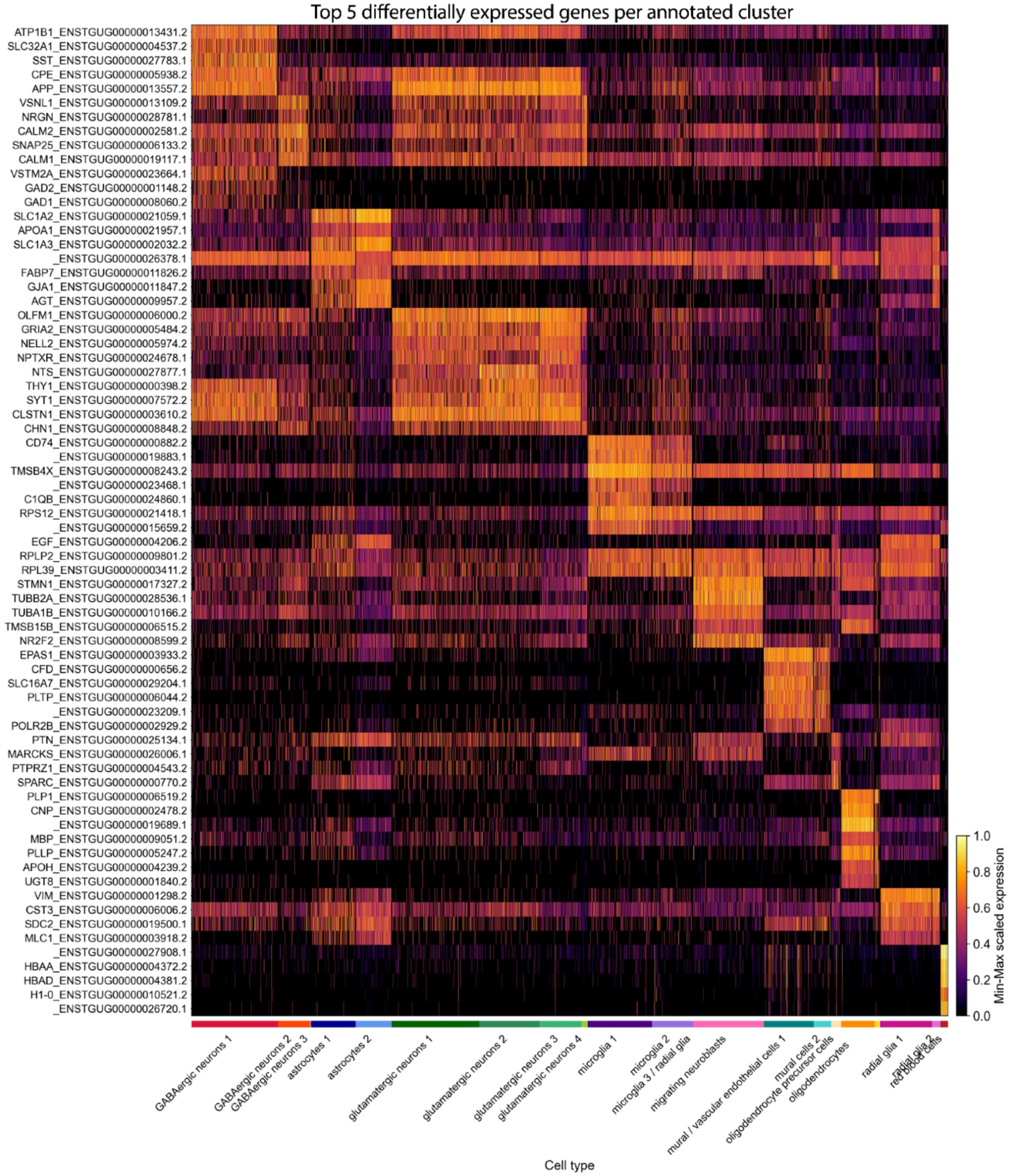
Sections were imaged under consistent conditions on a Nikon E600 microscope with a Lumina HR camera and imported into ImageJ for analysis. We quantified the expression level of the gene as measured by optical density and the number of cells expressing the gene per unit area, as previously described 54. Optical density was measured by taking the average pixel intensity of a 300x300 pixel square placed over the center of HVC. This value was normalized to the average background level of the tissue. To quantify the number of labeled cells, we established a threshold of expression that was 2.5x the background level. Binary filters (Close-, Open) were applied and the number of particles in the same 300x300 pixel square was quantified.

Histology

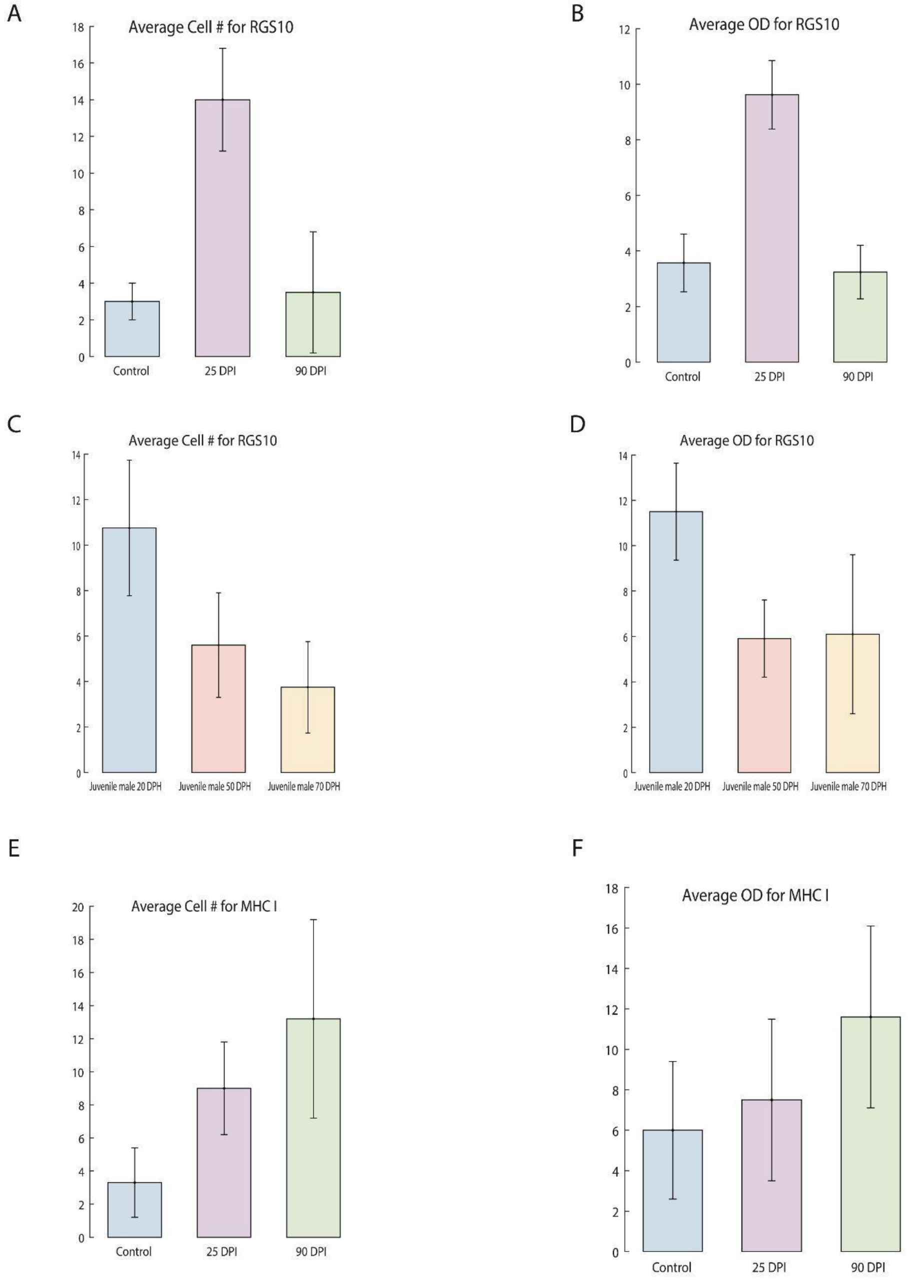
After cardiac perfusion with room temperature 3.2% PFA in 1xPBS we let the brains fix for 2-4 hours at room temperature. After each hemisphere of the brain was sectioned sagittally with a vibratome at 70-100 µm thickness. The brain slices containing HVC were collected and incubated at 4 C overnight with the primary rabbit anti-GFP (AB3080P, EMD Milipore) (blocked in 10% donkey serum in 0.2% Triton 1xPBS). On the second day, the brains were washed in 0.05% Triton 1xPBS and incubated for 2 hours in the dark at room temperature in the secondary goat anti-rabbit 488 (ab150077). Next, the brain slices were washed and mounted in Fluoromount (Sigma). Confocal images were taken with the LSM800.

Data and Code Availability

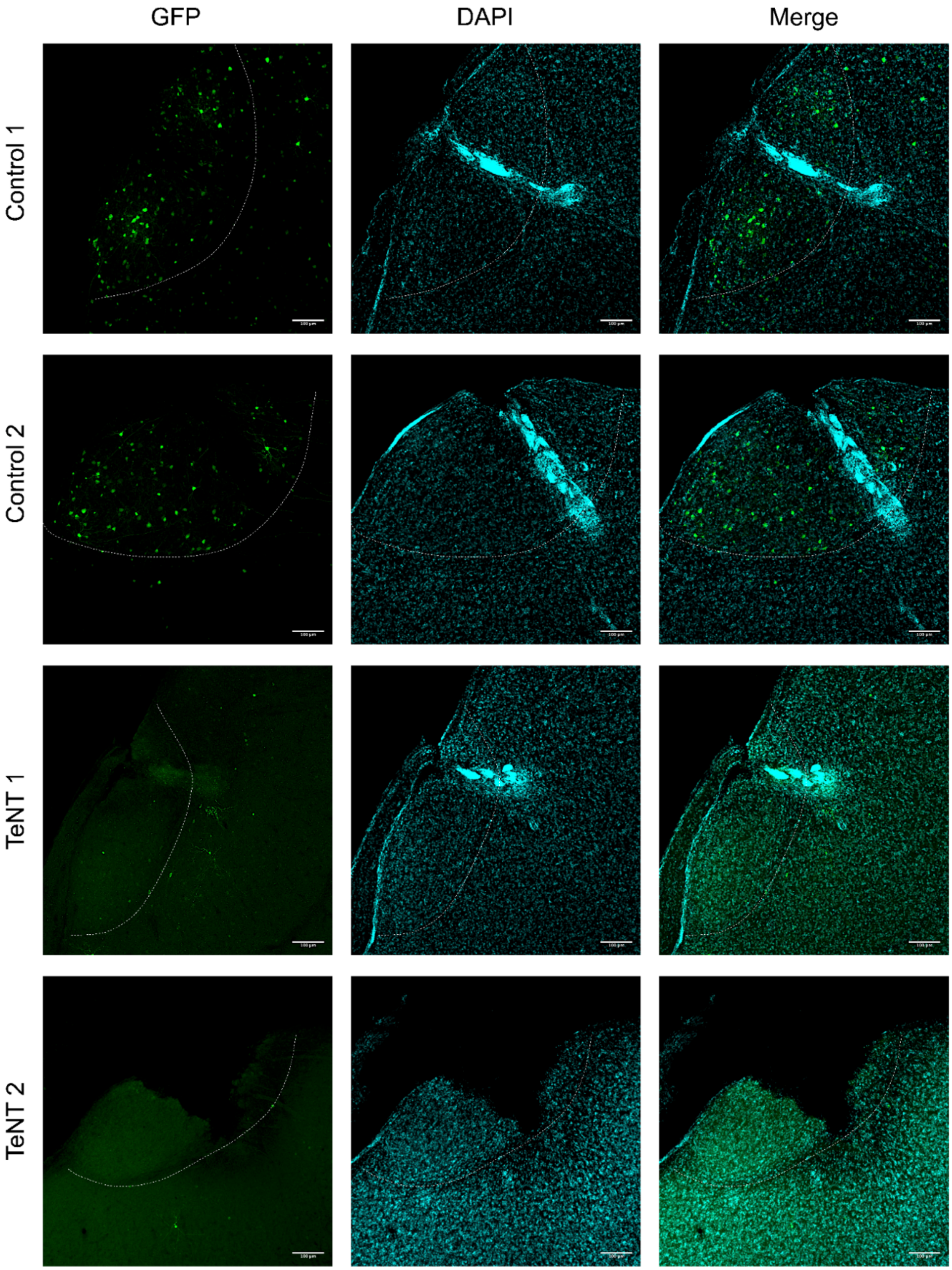
Data generated in this study have been deposited in Caltech DATA and can be found at the following DOIs: https://doi.org/10.22002/ednra-nn006 and https://doi.org/10.22002/3ta8v-gj982. Please do not hesitate to contact the authors for data or code requests. The code used for the analysis of the single-cell RNA sequencing data can be found here: https://github.com/lauraluebbert/TL_2023. The code used for the analysis of the chronic electrophysiology data can be found here: https://github.com/jordan-feldman/Torok2023-ephys.



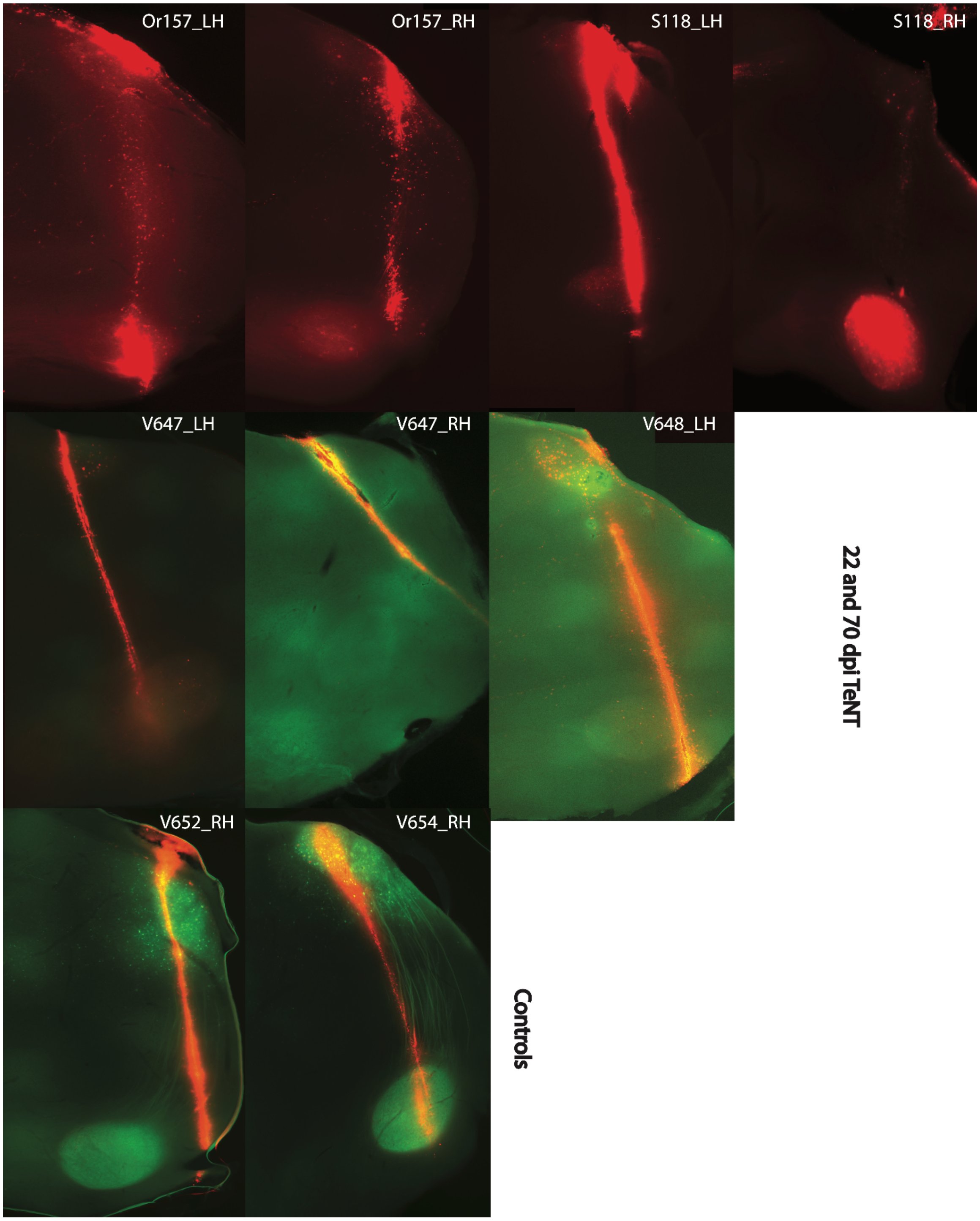
Supplementary Figure 4.1 Heatmap of top 5 differentially expressed genes per annotated cell type/cluster obtained by single-cell RNA sequencing of HVC from control and TeNT-treated birds at 25 dpi. Differentially expressed genes between clusters were identified using Scanpy's rank_genes_groups (p values were computed using a t-test and were adjusted with the Bonferroni method for multiple testing. They were then confirmed by comparison to p values generated with the nonparametric Wilcoxon test with Bonferroni correction). The heatmap depicts the min-max scaled expression for each gene.



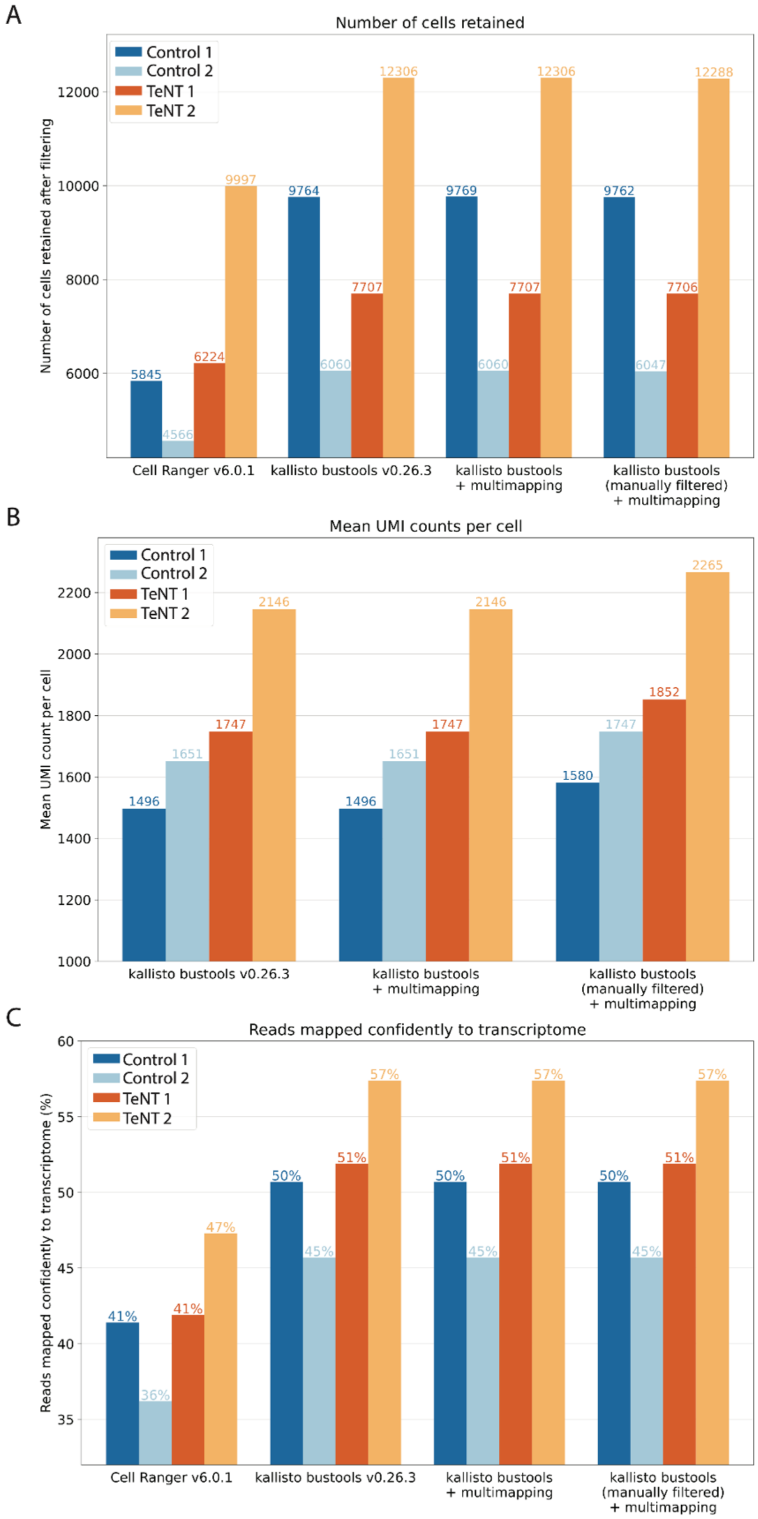
Supplementary Figure 4.2 Quantification of the *in situ* hybridization against microglia marker gene RGS10 in adult male control, TeNT-treated and juvenile male HVC; and MHC1 in adult male control, TeNT-treated animals. **A-B** Quantification of the *in situ* hybridization for RGS10 between control (n=4 animals) and TeNT-treated animals at 25 dpi (n=4) and 90 dpi (n=4). **C-D** Quantification of the *in situ* hybridization for RGS10 between juvenile males at 20, 50, and 70 days post-hatching (dph) (n=4). **E-F** Quantification of the *in situ* hybridization for MHC1 between control (n=4) and TeNT-treated animals at 25 (n=4) and 90 dpi (n=4). Error bars represent standard deviation.



Supplementary Figure 4.3 Histology of electrode array location in HVC in the chronically implanted animals. The white dotted line outlines HVC. Some sections display missing tissue due to the removal of the electrodes after perfusion of the animals. The stronger cyan signal indicates glial scar formation around the electrode array, which provides an approximation of the location of the electrodes.



Supplementary Figure 4.4 Histology to confirm the high-density silicone electrode location in the acute head-fixed animal recordings. The red trace represents the electrode location. The green trace represents the second electrode location in animals that were recorded twice, 40 days apart. The white labels represent the animal IDs. “LH” and “RH” stands for left and right hemisphere, respectively.



Supplementary Figure 4.5 Comparison of different pre-processing methods for the HVC single-cell RNA sequencing datasets. **A** Number of cells retained after quality control for each dataset and alignment method. **B** Mean UMI counts per cell for each dataset and pre-processing method. **C** Percentage of reads confidently mapped to transcriptome for each pre-processing method.

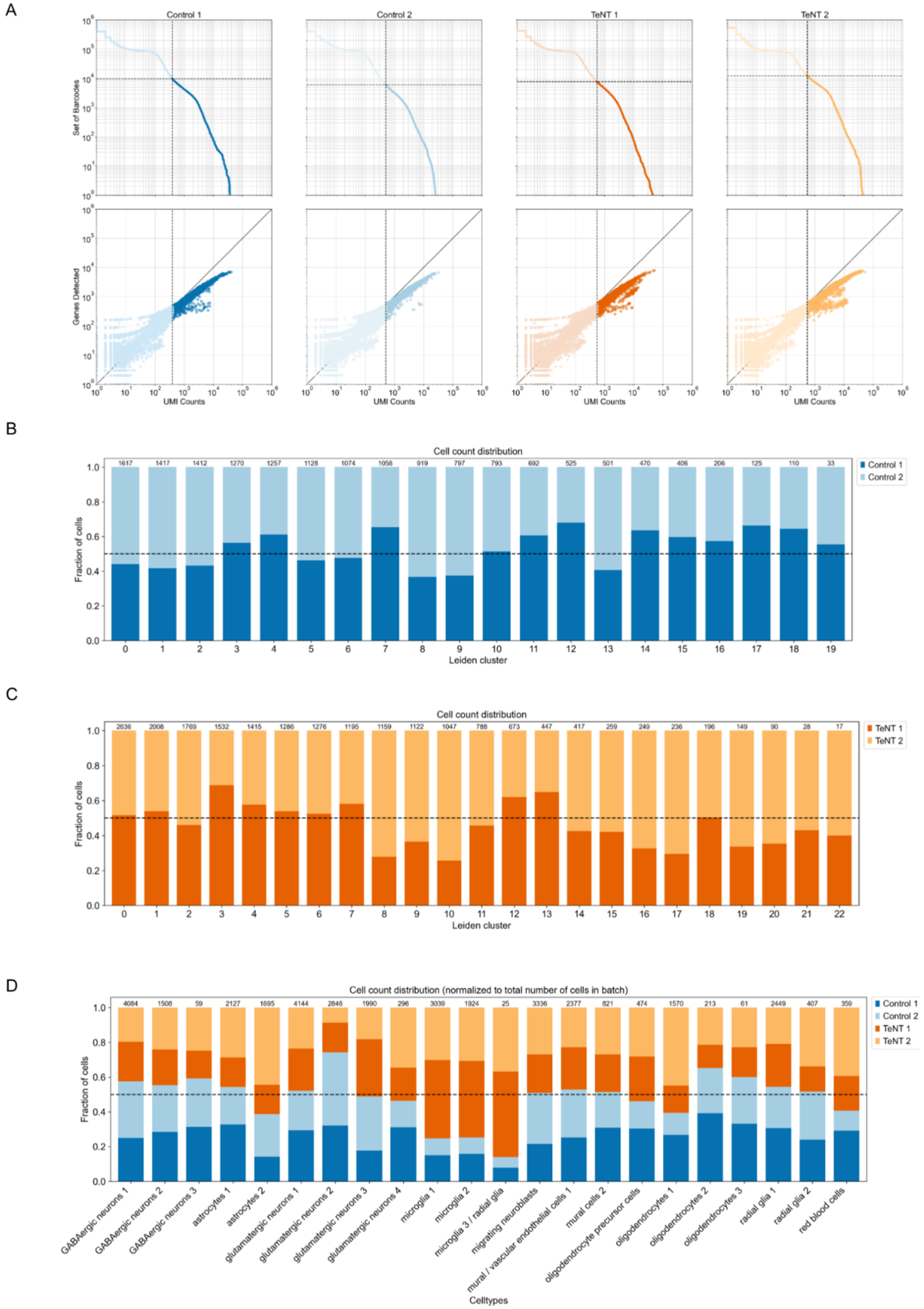


Figure legend on next page.

Supplementary Figure 4.6 Quality control of the single-cell RNA sequencing HVC datasets from control and TeTN-treated animals at 25 days post-injection (dpi). **A** “Knee plots” showing the set of barcodes (top row) and number of genes detected (bottom row) over UMI counts. The dashed lines depict the quality filtering cutoff. **B-C** Barplot depicting the fraction of cells from each replicate per cluster for control (B) and TeNT (C), normalized (by dividing) to the total number of cells in each replicate. Control and TeNT datasets were clustered separately using the Leiden algorithm. The equal distribution of replicates across the clusters suggests that technical effects do not dominate the clusters. Thus, we did not perform batch correction. The numbers on top of the bars indicate the total number of cells in each cluster. **D** Barplot depicting the fraction of cells from each dataset in the cell type clusters obtained after jointly clustering the control and TeNT datasets. The numbers on top of the bars indicate the total number of cells in each cluster.

Dataset (short name)	Species	Condition	Brain area	Replicate #	Technology	Pre-processing tool	# of cells retained after QC	Sequencing depth (number of reads processed)	Reads mapped confidently to transcriptome (%)	Mean UMI count per cell	Total UMI count
C1	Taeniopygia guttata	cag-neonGreen	HVC (both hemispheres)	1	10xv3	kallisto bustools	9,763	744,473,151	50.7	1580.8531	15,433,015
C2	Taeniopygia guttata	cag-neonGreen	HVC (both hemispheres)	2	10xv3	kallisto bustools	6,047	787,232,472	45.7	1747.639	10,568,132
E1	Taeniopygia guttata	dlx-TeNT-GFP	HVC (both hemispheres)	1	10xv3	kallisto bustools	7,706	867,768,600	51.9	1852.4215	14,274,784
E2	Taeniopygia guttata	dlx-TeNT-GFP	HVC (both hemispheres)	2	10xv3	kallisto bustools	12,288	810,253,355	57.4	2265.4258	27,837,586

Supplementary Table 4.1 Overview of single-cell RNA sequencing datasets.

References

1. Dehghani, N., Peyrache, A., Telenczuk, B., Le Van Quyen, M., Halgren, E., Cash, S.S., Hatsopoulos, N.G., and Destexhe, A. (2016). Dynamic Balance of Excitation and Inhibition in Human and Monkey Neocortex. *Sci. Rep.* 6, 23176.
2. Fritschy, J.-M. (2008). Epilepsy, E/I balance and GABA(A) receptor plasticity. Front. Mol. Neurosci. 1, 5.
3. Cossart, R., Bernard, C., and Ben-Ari, Y. (2005). Multiple facets of GABAergic neurons and synapses: multiple fates of GABA signalling in epilepsies. *Trends Neurosci.* 28, 108–115.
4. Nottebohm, F., Stokes, T.M., and Leonard, C.M. (1976). Central control of song in the canary, Serinus canarius. *J. Comp. Neurol.* 165, 457–486.
5. Kozhevnikov, A.A., and Fee, M.S. (2007). Singing-related activity of identified HVC neurons in the zebra finch. *J. Neurophysiol.* 97, 4271–4283.
6. Brainard, M.S., and Doupe, A.J. (2002). What songbirds teach us about learning. *Nature* 417, 351–358.
7. Vallentin, D., Kosche, G., Lipkind, D., and Long, M.A. (2016). Neural circuits. Inhibition protects acquired song segments during vocal learning in zebra finches. *Science* 351, 267–271.
8. Kosche, G., Vallentin, D., and Long, M.A. (2015). Interplay of inhibition and excitation shapes a premotor neural sequence. *J. Neurosci.* 35, 1217–1227.
9. Dimidschstein, J., Chen, Q., Tremblay, R., Rogers, S.L., Saldi, G.-A., Guo, L., Xu, Q., Liu, R., Lu, C., Chu, J., et al. (2016). A viral strategy for targeting and manipulating interneurons across vertebrate species. *Nat. Neurosci.* 19, 1743–1749.
10. Link, E., Edelmann, L., Chou, J.H., Binz, T., Yamasaki, S., Eisel, U., Baumert, M., Südhof, T.C., Niemann, H., and Jahn, R. (1992). Tetanus toxin action: inhibition of neurotransmitter release linked to synaptobrevin proteolysis. *Biochem. Biophys. Res. Commun.* 189, 1017–1023.
11. Vu, E.T., Mazurek, M.E., and Kuo, Y.C. (1994). Identification of a forebrain motor programming network for the learned song of zebra finches. *J. Neurosci.* 14, 6924–6934.
12. Yu, A.C., and Margoliash, D. (1996). Temporal hierarchical control of singing in birds. *Science* 273, 1871–1875.
13. Glaze, C.M., and Troyer, T.W. (2006). Temporal structure in zebra finch song: implications for motor coding. *J. Neurosci.* 26, 991–1005.
14. Brainard, M.S., and Doupe, A.J. (2000). Interruption of a basal ganglia–forebrain circuit prevents plasticity of learned vocalizations. *Nature* 404, 762–766.
15. Olveczky, B.P., Andalman, A.S., and Fee, M.S. (2005). Vocal experimentation in the juvenile songbird requires a basal ganglia circuit. *PLoS Biol.* 3, e153.
16. Kao, M.H., Doupe, A.J., and Brainard, M.S. (2005). Contributions of an avian basal ganglia–forebrain circuit to real-time modulation of song. *Nature* 433, 638–643.
17. Markowitz, J.E., Liberti, W.A., 3rd, Guitchounts, G., Velho, T., Lois, C., and Gardner, T.J. (2015). Mesoscopic patterns of neural activity support songbird cortical sequences. *PLoS Biol.* 13, e1002158.

18. Brown, D.E., 2nd, Chavez, J.I., Nguyen, D.H., Kadwory, A., Voytek, B., Arneodo, E.M., Gentner, T.Q., and Gilja, V. (2021). Local field potentials in a pre-motor region predict learned vocal sequences. *PLoS Comput. Biol.* 17, e1008100.
19. Crandall, S.R., Adam, M., Kinnischtzke, A.K., and Nick, T.A. (2007). HVC neural sleep activity increases with development and parallels nightly changes in song behavior. *J. Neurophysiol.* 98, 232–240.
20. Dave, A.S., and Margoliash, D. (2000). Song replay during sleep and computational rules for sensorimotor vocal learning. *Science* 290, 812–816.
21. Elmaleh, M., Kranz, D., Asensio, A.C., Moll, F.W., and Long, M.A. (2021). Sleep replay reveals premotor circuit structure for a skilled behavior. *Neuron* 109, 3851–3861.e4.
22. Hahnloser, R.H.R., Kozhevnikov, A.A., and Fee, M.S. (2006). Sleep-related neural activity in a premotor and a basal-ganglia pathway of the songbird. *J. Neurophysiol.* 96, 794–812.
23. Shank, S.S., and Margoliash, D. (2009). Sleep and sensorimotor integration during early vocal learning in a songbird. *Nature* 458, 73–77.
24. Wang, B., Torok, Z., Duffy, A., Bell, D., Wongso, S., Velho, T., Fairhall, A., and Lois, C. (2022). Unsupervised Restoration of a Complex Learned Behavior After Large-Scale Neuronal Perturbation. *bioRxiv*, 2022.09.09.507372. 10.1101/2022.09.09.507372.
25. Jun, J.J., Steinmetz, N.A., Siegle, J.H., Denman, D.J., Bauza, M., Barbarits, B., Lee, A.K., Anastassiou, C.A., Andrei, A., Aydın, Ç., et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature* 551, 232–236.
26. Steinmetz, N.A., Koch, C., Harris, K.D., and Carandini, M. (2018). Challenges and opportunities for large-scale electrophysiology with Neuropixels probes. *Curr. Opin. Neurobiol.* 50, 92–100.
27. Fisher, R.S., Scharfman, H.E., and deCurtis, M. (2014). How can we identify ictal and interictal abnormal activity? *Adv. Exp. Med. Biol.* 813, 3–23.
28. Lubenov, E.V., and Siapas, A.G. (2009). Hippocampal theta oscillations are travelling waves. *Nature* 459, 534–539.
29. Buzsáki, G. (1986). Hippocampal sharp waves: their origin and significance. *Brain Res.* 398, 242–252.
30. Hulse, B.K., Lubenov, E.V., and Siapas, A.G. (2017). Brain State Dependence of Hippocampal Subthreshold Activity in Awake Mice. *Cell Rep.* 18, 136–147.
31. Joo, H.R., and Frank, L.M. (2018). The hippocampal sharp wave–ripple in memory retrieval for immediate use and consolidation. *Nat. Rev. Neurosci.* 19, 744–757.
32. Lenz, K.M., and Nelson, L.H. (2018). Microglia and Beyond: Innate Immune Cells As Regulators of Brain Development and Behavioral Function. *Front. Immunol.* 9, 698.
33. Li, Q., and Barres, B.A. (2018). Microglia and macrophages in brain homeostasis and disease. *Nat. Rev. Immunol.* 18, 225–242.
34. Thion, M.S., Ginhoux, F., and Garel, S. (2018). Microglia and early brain development: An intimate journey. *Science* 362, 185–189.
35. Parkhurst, C.N., Yang, G., Ninan, I., Savas, J.N., Yates, J.R., 3rd, Lafaille, J.J., Hempstead, B.L., Littman, D.R., and Gan, W.-B. (2013). Microglia promote learning-

dependent synapse formation through brain-derived neurotrophic factor. *Cell* 155, 1596–1609.

36. Simpson, E. (1988). Function of the MHC. *Immunol. Suppl.* 1, 27–30.
37. Elmer, B.M., and McAllister, A.K. (2012). Major histocompatibility complex class I proteins in brain development and plasticity. *Trends Neurosci.* 35, 660–670.
38. Chacon, M.A., and Boulanger, L.M. (2013). MHC class I protein is expressed by neurons and neural progenitors in mid-gestation mouse brain. *Mol. Cell. Neurosci.* 52, 117–127.
39. Shatz, C.J. (2009). MHC class I: an unexpected role in neuronal plasticity. *Neuron* 64, 40–45.
40. Lazarczyk, M.J., Kemmler, J.E., Eyford, B.A., Short, J.A., Varghese, M., Sowa, A., Dickstein, D.R., Yuk, F.J., Puri, R., Biron, K.E., et al. (2016). Major Histocompatibility Complex class I proteins are critical for maintaining neuronal structural complexity in the aging brain. *Sci. Rep.* 6, 26199.
41. Tchernichovski, O., Nottebohm, F., Ho, C.E., Pesaran, B., and Mitra, P.P. (2000). A procedure for an automated measurement of song similarity. *Anim. Behav.* 59, 1167–1176.
42. Goffinet, J., Brudner, S., Mooney, R., and Pearson, J. (2021). Low-dimensional learned feature spaces quantify individual and group differences in vocal repertoires. *Elife* 10. 10.7554/eLife.67855.
43. Sainburg, T., Thielk, M., and Gentner, T.Q. (2020). Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS Comput. Biol.* 16, e1008228.
44. Yeganegi, H., Luksch, H., and Ondracek, J.M. (2019). Hippocampal-like network dynamics underlie avian sharp wave-ripples. *bioRxiv*, 825075. 10.1101/825075.
45. Shan, K.Q., Lubenov, E.V., and Siapas, A.G. (2017). Model-based spike sorting with a mixture of drifting t-distributions. *J. Neurosci.* Methods 288, 82–98.
46. Pachitariu, M., Steinmetz, N., Kadir, S., Carandini, M., and Harris, K.D. (2016). Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *bioRxiv*, 061481. 10.1101/061481.
47. Pool, A.-H., Wang, T., Stafford, D.A., Chance, R.K., Lee, S., Ngai, J., and Oka, Y. (2020). The cellular basis of distinct thirst modalities. *Nature* 588, 112–117.
48. Melsted, P., Sina Booeshaghi, A., Gao, F., Beltrame, E., Lu, L., Hjorleifsson, K.E., Gehring, J., and Pachter, L. (2019). Modular and efficient pre-processing of single-cell RNA-seq. *bioRxiv*, 673285. 10.1101/673285.
49. Wolf, F.A., Angerer, P., and Theis, F.J. (2018). SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* 19, 15.
50. Luebbert, L., and Pachter, L. (2023). Efficient querying of genomic reference databases with gget. *Bioinformatics* 39. 10.1093/bioinformatics/btac836.
51. Traag, V.A., Waltman, L., and van Eck, N.J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* 9, 5233.
52. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nat. Methods* 15, 1053–1058.

53. Colquitt, B.M., Merullo, D.P., Konopka, G., Roberts, T.F., and Brainard, M.S. (2021). Cellular transcriptomics reveals evolutionary identities of songbird vocal circuits. *Science* 371. 10.1126/science.abd9704.
54. Zemel, B.M., Nevue, A.A., Dagostin, A., Lovell, P.V., Mello, C.V., and von Gersdorff, H. (2021). Resurgent Na+ currents promote ultrafast spiking in projection neurons that drive fine motor control. *Nat. Commun.* 12, 6762.
55. Carleton, J.B., Lovell, P.V., McHugh, A., Marzulla, T., Horback, K.L., and Mello, C.V. (2014). An optimized protocol for high-throughput in situ hybridization of zebra finch brain. *Cold Spring Harb. Protoc.* 2014, 1249–1258.
56. Nevue, A.A., Lovell, P.V., Wirthlin, M., and Mello, C.V. (2020). Molecular specializations of deep cortical layer analogs in songbirds. *Sci. Rep.* 10, 18767.

Chapter 5

TRANSCRIPTOMICS IN HEALTHCARE – PART I

PSCA-CAR T cells for Metastatic Castration-resistant Prostate Cancer: A Phase 1 Trial

Preamble

This chapter describes the results obtained through single-cell RNA sequencing of the CAR T product, peripheral blood, and solid tumor tissue derived from 12 prostate cancer patients at varying time points of CAR T therapy using two different sequencing technologies, single-cell gene expression and V(D)J immune repertoire sequencing, resulting in a total of 64 multiplexed datasets and over 25 billion sequencing reads which I analyzed in parallel, revealing significant differences in the immune landscape dynamics and identifying relevant time points for clonotype expansion. Below, I am only including the relevant single-cell RNA sequencing results and methods from the published paper.

Tanya B. Dorff, M. Suzette Blanchard, Lauren N. Adkins, **Laura Luebbert,** Neena Leggett, Stephanie N. Shishido, Alan Macias, Marissa Del Real, Gaurav Dhapola, Colt Egelston, John P. Murad, Reginaldo Rosa, Jinny Paul, Ammar Chaudhry, Hripsime Martirosyan, Ethan Gerdts, Jamie R. Wagner, Tracey Stiller, Dileshni Tilakawardane, Sumanta Pal, Robert E. Reiter, Catalina Martinez, Elizabeth L. Budde, Massimo D'Apuzzo, Peter Kuhn, Lior Pachter, Stephen J. Forman, Saul J. Priceman (2024). PSCA-CAR T cell therapy for metastatic castration-resistant prostate cancer: a phase 1 trial. *Under review.*

Summary

Despite recent therapeutic advances, metastatic castration-resistant prostate cancer (mCRPC) remains lethal. Chimeric antigen receptor (CAR) T cell therapies have demonstrated durable remissions in hematological malignancies and are of interest for patients with mCRPC. We report results from a phase 1, first-in-human study of prostate stem cell antigen (PSCA)-directed CAR T cells in patients with mCRPC screened for tumor PSCA expression. The starting dose level was 100 million (M) CAR T cells without lymphodepletion (LD), followed by incorporation of LD with 100M CAR T cells. The primary endpoints were safety and dose-limiting toxicities (DLTs). No DLTs were observed at DL1, with a DLT of grade 3 cystitis encountered at DL2, resulting in addition of a new cohort using a reduced LD regimen + 100M CAR T cells (DL3). No DLTs were observed in DL3. Cytokine release syndrome (CRS) of grade 1 or 2 occurred in 5 of 14 treated patients. PSA declines (>30%) occurred in 4 of 14 patients, as well as radiographic improvements. Dynamic changes indicating activation of peripheral blood endogenous and CAR T cell subsets, TCR repertoire diversity, and changes in the tumor immune microenvironment were observed in a subset of patients. Limited persistence of CAR T cells was observed beyond 28 days post infusion. In summary, CAR T cells targeting PSCA

demonstrate bioactivity at a single dose of 100M, supporting future clinical studies evaluating multiple infusions to achieve higher total dose and combinatorial approaches are needed to improve durable therapeutic outcomes.

Introduction

Metastatic castration-resistant prostate cancer (mCRPC) is a lethal disease, causing more than 30,000 deaths in American men each year (1). Immunotherapy has largely been unsuccessful; both vaccine-based strategies such as GVAX and Prost-VAC (2,3) and immune checkpoint inhibition with CTLA-4 and PD-1 inhibitors (4,5) have shown limited activity. The only immunotherapy proven to prolong survival in mCRPC is sipuleucel-T, which is an autologous cellular immunotherapy with ex-vivo incubation of dendritic cells leading to activation against prostate acid phosphatase (6). However, significant improvements are needed for immunotherapies to effectively target mCRPC.

Reasons for lack of immunotherapy response in prostate cancer are multi-fold, including strong immunosuppression in advanced prostate cancer (7) that limits both trafficking and effector T cell function in the local tumor microenvironment. Despite this, there are unique tumor-associated antigens in mCRPC which are commonly and robustly expressed including prostate stem cell antigen (PSCA) and prostate specific membrane antigen (PSMA), which could be leveraged as targets for powerful cellular immunotherapy modalities. The dramatic successes of chimeric antigen receptor (CAR) T cell therapies in hematological malignancies have inspired the clinical development of CAR T cell therapies for the treatment of mCRPC.

PSCA is highly expressed in prostate cancer, and increases with advanced disease states, particularly in the setting of bone metastases (8). Using xenograft and syngeneic tumor models, we demonstrated safety and efficacy of second generation PSCA-CAR T cells with 4-1BB costimulation in eradicating bone metastatic prostate cancer (9). Here, we report results of our first-in-human phase 1 clinical trial to evaluate the safety and bioactivity of PSCA-CAR T cells in mCRPC patients.

Results

Clinical trial design and patient characteristics

City of Hope conducted a single center, first-in-human, phase 1 clinical trial to evaluate safety and bioactivity of PSCA-directed CAR T cells in patients with metastatic castration-resistant prostate cancer (NCT03873805). The primary endpoints were safety and dose-limiting toxicities (DLT). The secondary endpoints were persistence of CAR T cells to 28 days post infusion (defined as CAR T cells comprising at least 7.5 copies/µg of DNA of total CD3 cells), expansion of CAR T cells (Max log10 copies/µg of genomic DNA, disease response (PSA decline, RECIST) and survival described as percent of participants alive at 6 months. Exploratory endpoints were phenotypes and frequencies of immune cell subsets in the peripheral blood pre- and post-therapy, serum cytokine profile before and after CAR T infusion to assess potential CRS toxicity and CAR T cell effector function, phenotype of tumor-infiltrating lymphocytes, gene expression (by RNA-seq) of CTCs, cfDNA in

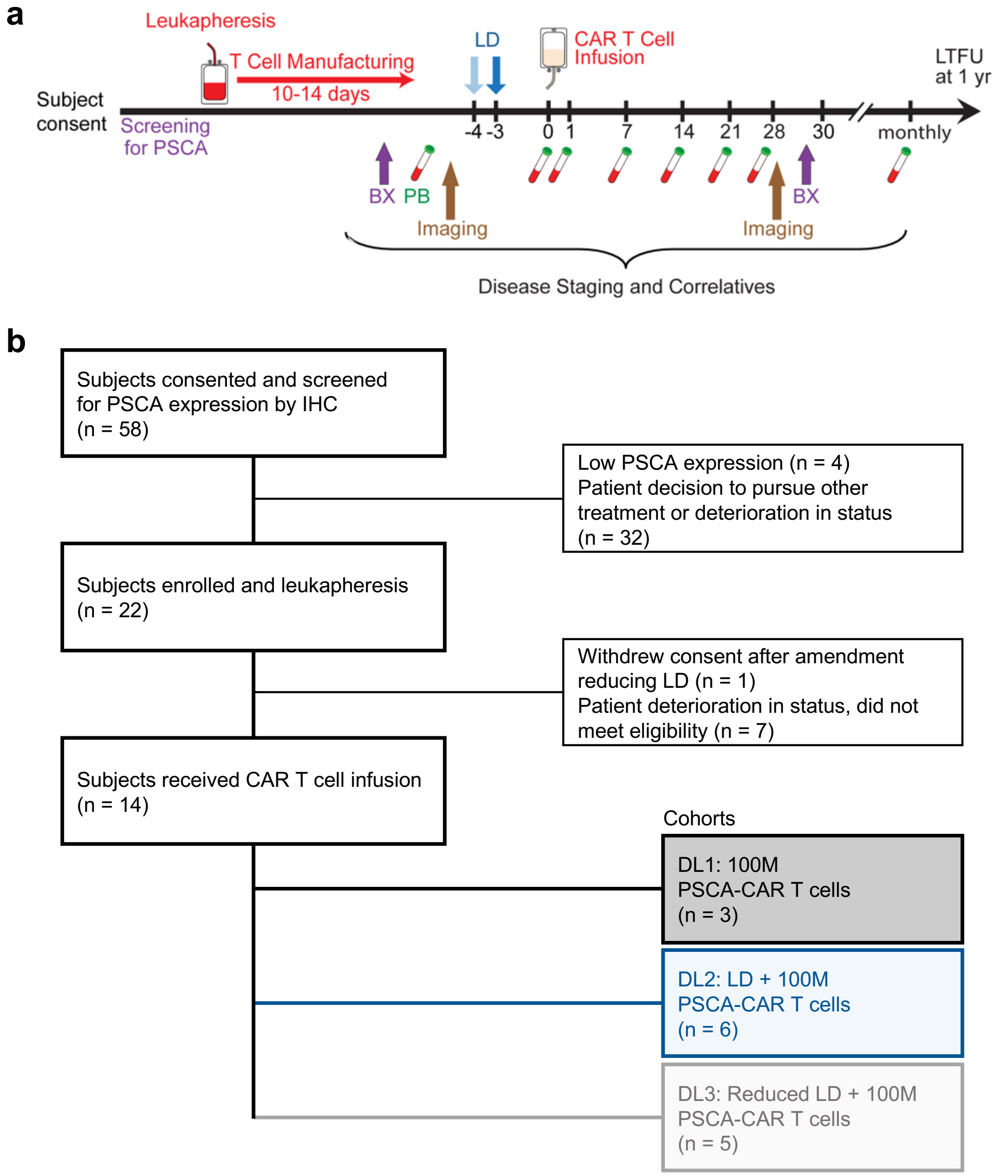

a
Leukapheresis
T Cell Manufacturing
10-14 days
LD
CAR T Cell Infusion
LTFU at 1 yr
Subject consent
Screening for PSCA
-4
-3
0
1
7
14
21
28
30
monthly
BX
PB
Imaging
Imaging
BX
Disease Staging and Correlatives
b
Subjects consented and screened for PSCA expression by IHC (n = 58)
Low PSCA expression (n = 4)
Patient decision to pursue other treatment or deterioration in status (n = 32)
Subjects enrolled and leukapheresis (n = 22)
Withdrew consent after amendment reducing LD (n = 1)
Patient deterioration in status, did not meet eligibility (n = 7)
Subjects received CAR T cell infusion (n = 14)
Cohorts
DL1: 100M PSCA-CAR T cells (n = 3)
DL2: LD + 100M PSCA-CAR T cells (n = 6)
DL3: Reduced LD + 100M PSCA-CAR T cells (n = 5)

Figure 5.1 Clinical trial design and CONSORT diagram. **(a)** Illustration of clinical trial design including subject screening, leukapheresis, PSCA-CAR T cell manufacturing, pre-infusion biopsy (BX), peripheral blood (PB) sample collection prior to lymphodepletion (LD), bone scan and CT imaging, Flu/Cy LD, PSCA-CAR T cell infusion, serial PB sample collection timepoints from day 0 to day 28, post-infusion bone scan and CT imaging, post-infusion BX, and long-term follow up (LTFU). **(b)** CONSORT diagram detailing subjects consented and screened for PSCA expression by immunohistochemistry (IHC) (n = 58), subjects enrolled and leukapheresis (n = 22), subjects received CAR T cell infusion (n = 14). Dose level (DL) cohorts including DL1 (100 million (M) PSCA-CAR T cells, n = 3), DL2 (Flu/Cy LD + 100M PSCA-CAR T cells, n = 6), and DL3 (Reduced Flu/Cy LD + 100M PSCA-CAR T cells, n = 5).

peripheral blood by whole exome sequencing, and CAR immunogenicity (anti-PSCA-CAR antibodies).

The clinical trial design is summarized in Figure 5.1a. Fifty-eight subjects were screened for PSCA expression by immunohistochemistry, twenty-two subjects underwent leukapheresis and CAR T cell manufacturing, and fourteen subjects were treated from August 2019 to July 2022 (Figure 5.1b CONSORT diagram). The first subject was pre-screened (tissue testing) 5/23/19, first subject enrolled (leukapheresis) 7/30/19, and last subject treated (CAR T cell infusion) 7/25/22; the trial is closed. The median age of subjects on study was 62 for dose level (DL)1, 70 for DL2, and 69 for DL3. All subjects received prior androgen receptor signaling inhibitors, either enzalutamide (71%), abiraterone (79%), or both (64%) and a majority of patients received cabazitaxel (57%), docetaxel (86%), or both (57%) prior to CAR T cell infusion. Baseline PSA (median) ranged from 16.5 to 235.3.

CAR T cell product manufacturing and characterization

The PSCA-CAR construct comprised the anti-PSCA humanized scFv (A11 clone), □CH2 extracellular spacer, CD4 transmembrane domain, 4-1BB intracellular co-stimulatory domain, and CD3γ cytolytic domain as previously published (9). Briefly, CAR T cell manufacturing included depletion of CD14+ and CD25+ cells, CD3/28 bead stimulation, transduction with lentivirus at multiplicity of infection of 0.1, removal of beads at day 7-9, followed by expansion for a total of 12-17 days in IL-2 and IL-15 cytokines. There were no manufacturing failures, with a median CAR percentage of 86.8% in the final released product. Thawed products were characterized by flow cytometry for expression of CD4/CD8 , CD19 (for CD19t transduction marker) expression, and Fc (PSCA-CAR) expression, as well as T cell subsets demonstrating a dominant Tcm/Tem phenotype. Two products fell outside the pre-specified woodchuck post-transcriptional regulatory element (WPRE) copy number (<5), and FDA approval was granted to proceed with infusion. Median time from leukapheresis to infusion of the product was 73 days (range 34 to 182); delays were primarily due to protocol mandated holds on accrual during toxicity assessments and protocol amendments, waiting for confirmatory PSCA staining from on-study biopsies, as well as seeking regulatory approval for the use of product out of parameters (as specified above). Six patients received bridging therapy: cabazitaxel (4), cabazitaxel + carboplatin (1), enzalutamide (1).

Treatment response

PSA declines from before treatment to day 28 after CAR T cell infusion were seen in 1 of 3 subjects in DL1, 3 of 6 subjects in DL2, and 3 of 5 subjects in DL3. Waterfall plot of the maximum PSA change from before CAR T cell infusion to day 28 shows 4 of 14 subjects with PSA declines >30% (Figure 5.2a). Of these, only 1 subject maintained PSA decline >30% beyond 28 days. In DL1, 1 of 3 subjects treated experienced a transient PSA response; notably this subject had evidence of early neuroendocrine (NE) expression in the on-study biopsy but still retained strong PSCA expression, and RECIST response was PD. Post-treatment biopsy revealed further NE transformation (data not shown). The first

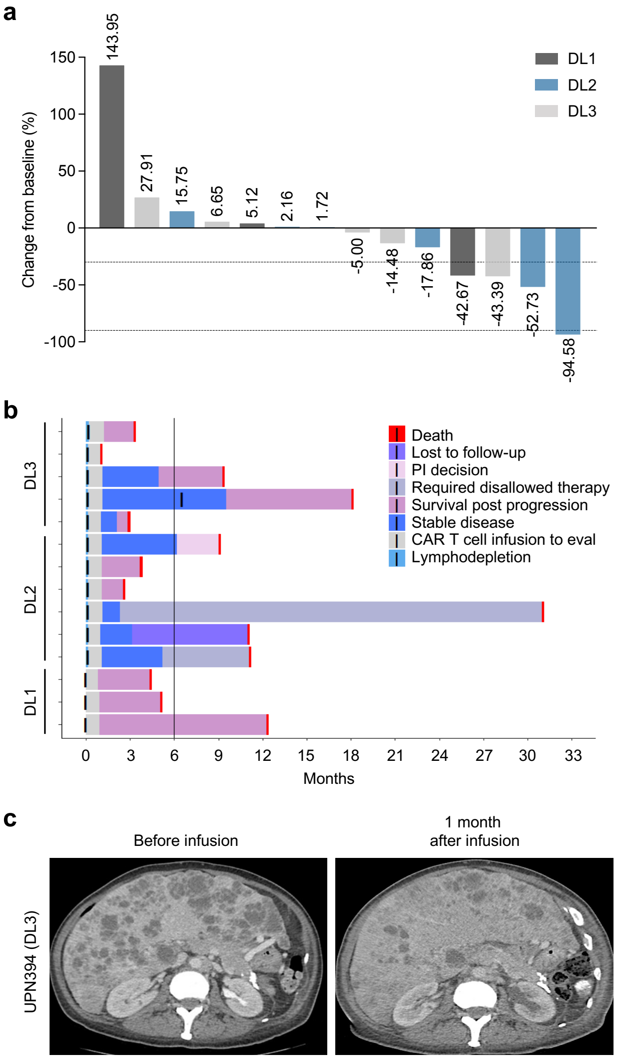


Figure 5.2 Treatment response following PSCA-CAR T cell infusion. **(a)** PSA waterfall plot showing best PSA response in the 28 days following CAR T cell infusion at each dose level (DL). **(b)** Swimmer's plot depicting response to treatment and follow up for each subject on study. **(c)** Computed tomography (CT) scan of a patient (UPN394) in DL3 showing liver metastases before infusion and disease response 1 month after infusion of PSCA-CAR T cells.

subject treated in DL2 (with lymphodepletion) achieved a >90% PSA decline in the first 28 days post CAR T cell infusion. The response in this subject is characterized in greater detail below.

Rates of stable disease by RECIST were DL1: 0%, DL2 67%, and DL3 60%. Swimmer plots for treated subjects are shown in Figure 5.2b, with a 33%, 67%, and 40% 6-month survival rate in DL1, DL2, and DL3, respectively. The first subject treated in DL3 achieved radiographic improvement in liver metastatic burden but did not achieve PSA response (Figure 5.2c). One subject with bone only disease who exhibited stable disease in DL3 requested and received a 2nd infusion of 100M CAR T cells about 6 months following initial infusion. He experienced transient relief of cancer-related pain after the 2nd infusion.

We also evaluated treatment response by circulating tumor cell (CTC) quantification in the peripheral blood of treated subjects on study using high-definition single cell analysis (HDSCA) (10). Cytokeratin (CK)-positive cells were detected in the peripheral blood of 100% of treated subjects. Overall, there were marked declines in mean CK+ cells from baseline to 28 days after CAR T cell infusion in both of the LD cohorts (DL2 and DL3), but not in DL1.

Somatic DNA sequencing results were available for 8 subjects and 2 subjects had germline testing results (no overlap between somatic and germline tested patients). Of the 8 with somatic testing, the highest tumor mutational burden was 10.5, all others were deemed low.

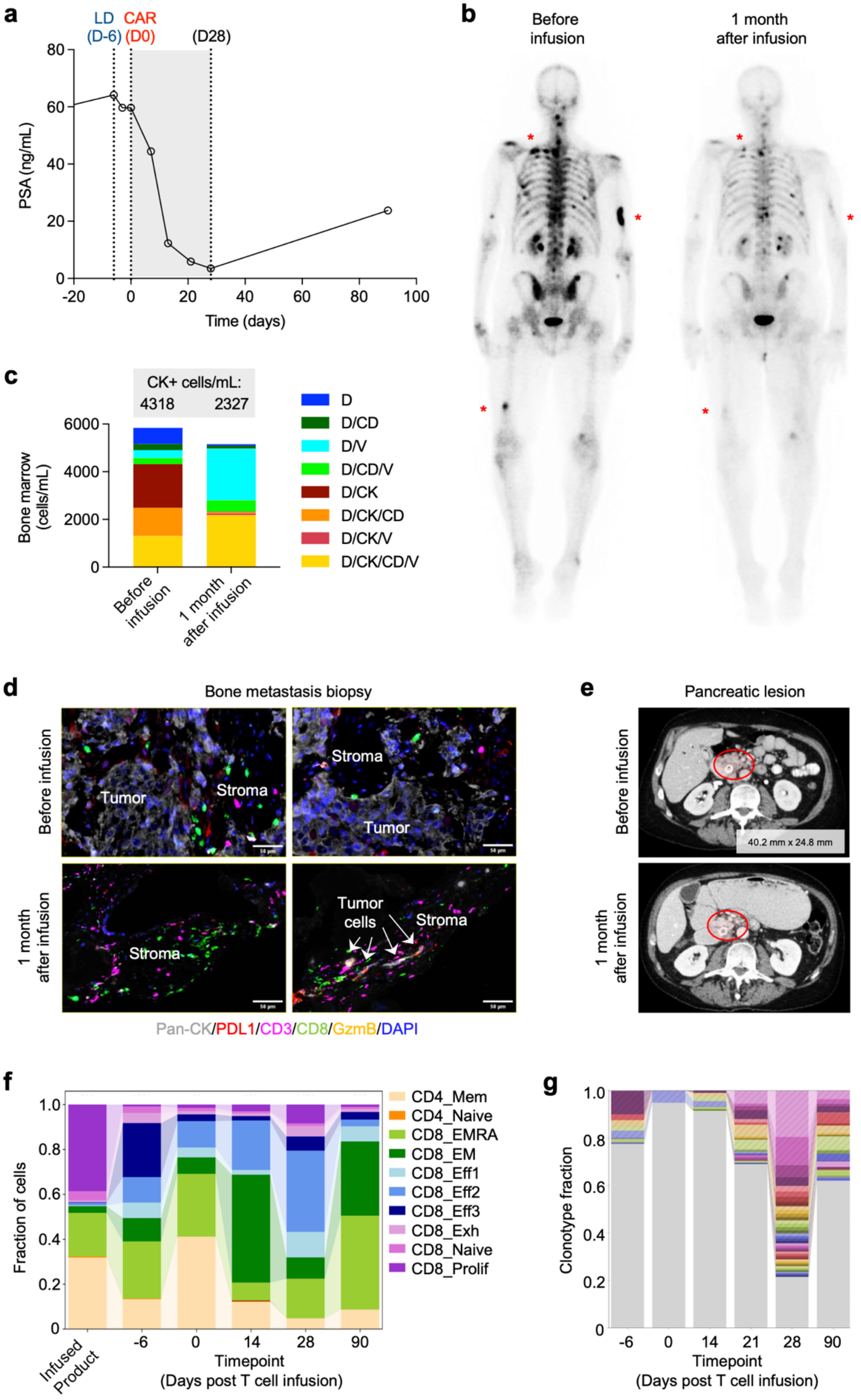

a
LD
(D-6)
CAR
(D0)
(D28)
PSA (ng/mL)
80
60
40
20
0
-20
0
20
40
60
80
100
Time (days)
b
Before
infusion
1 month
after infusion
*
c
CK+ cells/mL:
4318
2327
Bone marrow
(cells/mL)
6000
4000
2000
0
Before
infusion
1 month
after infusion
D
D/CD
D/V
D/CD/V
D/CK
D/CK/CD
D/CK/V
D/CK/CD/V
d
Bone metastasis biopsy
Before infusion
1 month
after infusion
Tumor
Stroma
Stroma
Tumor
Stroma
Tumor
cells
Stroma
Pan-CK/PDL1/CD3/CD8/GzmB/DAPI
e
Pancreatic lesion
Before infusion
1 month
after infusion
40.2 mm x 24.8 mm
f
Fraction of cells
1.0
0.8
0.6
0.4
0.2
0
Infused
Product
-6
0
14
28
90
Timepoint
(Days post T cell infusion)
CD4_Mem
CD4_Naive
CD8_EMRA
CD8_EM
CD8_Eff1
CD8_Eff2
CD8_Eff3
CD8_Exh
CD8_Naive
CD8_Prolif
g
Clonotype fraction
1.0
0.8
0.6
0.4
0.2
0
-6
0
14
21
28
90
Timepoint
(Days post T cell infusion)

Figure 5.3 Patient with biochemical and radiographic response with associated immune landscape changes. **(a)** PSA response in UPN388 on DL2 before and through the 28 days following PSCA-CAR T cell infusion and at day 90. **(b)** bone scintigraphy (anterior-posterior view) for bone metastases detection before and 1 month after PSCA-CAR T cell infusion in the same patient. Red asterisks denote representative bone metastases. **(c)** High-definition single cell analysis (HDSCA) of circulating tumor cells (CTCs) in the bone marrow before and 1 month after infusion of PSCA-CAR T cells. Quantification of CK+ cells per mL is shown in grey box. **(d)** Immunofluorescence images of bone metastasis biopsy samples from before (top) and 1 month after PSCA-CAR T cell infusion (bottom), evaluating expression of pan-cytokeratin (pan-CK) (tumor cells), PDL1, CD3 (T cells), CD8 (effector cells), and Granzyme B (GzmB). Indicated areas of tumor and stromal regions, and arrows indicate residual tumor cells in post-infusion sample. Images shown are representative of the whole evaluable tissue region on slide. **(e)** Computed tomography (CT) scan of pancreatic lesion in UPN388 before and 1 month after PSCA-CAR T cell infusion. Red circles denote pancreatic lesion around stent. Measured size of lesion before infusion, 40.2 mm x 24.8 mm. Lesion regressed 1 month after infusion and was not measurable. **(f)** scRNAseq analysis of CD3+ T cell subsets in the infused product and in the peripheral blood T cells at indicated timepoints post-T cell infusion. **(g)** Single cell analysis of TCRa/b repertoire diversity in the peripheral blood T cells at indicated timepoints post-T cell infusion. Top 40 clonotypes with greatest fractions at day 28. Legend in Figure 5.4e.

PTEN loss was noted in 3 of these subjects, one of whom experienced the greatest PSA decline on study (Figure 5.2a). In DL3, the subject with radiographic improvement in liver metastases had a genomic alteration in CDK12, and he had progressed on prior immune checkpoint inhibitor therapy.

Patient with biochemical and radiographic response

One participant, a subject in DL2, experienced a >90% PSA decline following CAR T cell infusion, from 64.2 ng/mL before LD and CAR T cells to 3.5 ng/mL at day 28 after CAR T cell infusion (Figure 5.3a). Radiographic improvement was seen in this subject's soft tissue metastasis (Figure 5.3b) though RECIST assessment was SD due to the presence of bone metastases. Changes in serum cytokines in this patient demonstrate pronounced but transient induction of inflammatory factors, including IFNy, IL-6, GM-CSF, IP-10, and MIG. Serum chemistry showed mild increases in CRP (max 81 mg/L), ferritin (max 555 ng/mL), ALT/AST (<1.5 x ULN), LDH (max 365 U/L), and alkaline phosphatase (max 192) following CAR T cell infusion. This corresponded to grade 2 CRS with T max 39.1 on day 4, 38.8 on day 5 and tocilizumab was administered on day 6 due to persistent rigors without fever; all aforementioned labs subsequently trended down by day 21. CTC assessment with cytokeratin positivity were significantly reduced, both in bone marrow (Figure 5.3c) and in peripheral blood samples, from baseline to 28 days after CAR T infusion. The post-CAR T cell infusion bone metastasis biopsy showed reductions in PSCA+ disease, Ki67+ expression, along with greater infiltration of CD3+ and cytotoxic CD8+ T cells by immunofluorescence staining (Figure 5.3d). Few residual tumor cells in the post-treatment biopsy were observed and were associated with increased granzyme B+ and PD-L1+ areas, suggestive of an active anti-tumor immune response. Quantification of immunofluorescence staining showed increased CD8+ and PD-L1+ areas in this subject, with variable results from other subjects analyzed. Interestingly, UPN388 also had a biopsy proven prostate cancer metastasis in the pancreas which necessitated stent placement prior to study entry; this completely resolved after CAR T cell infusion (Figure 5.3e).

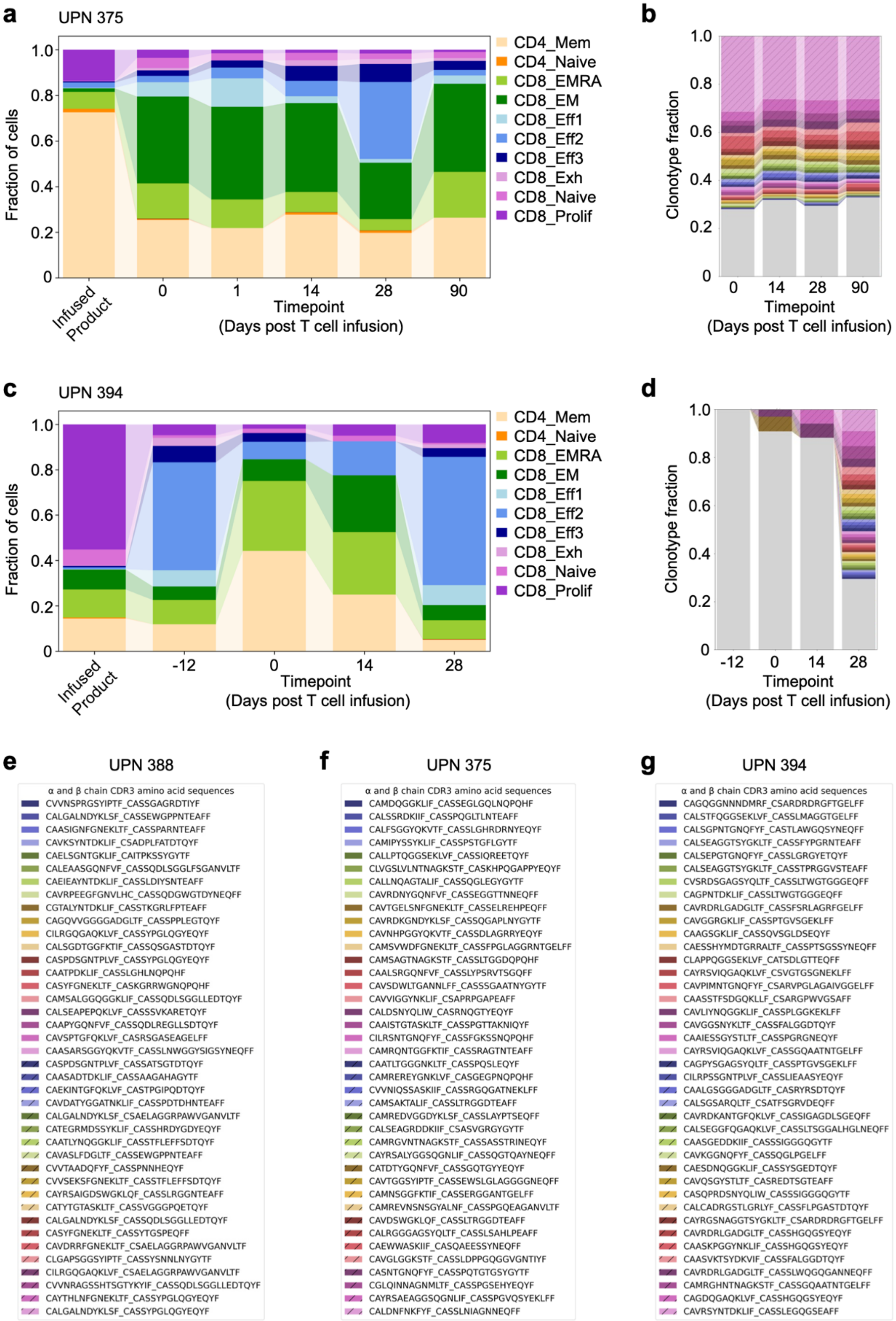

a
UPN 375
Fraction of cells
0
0.2
0.4
0.6
0.8
1.0
Infused Product
0
1
14
28
90
Timepoint
(Days post T cell infusion)
CD4_Mem
CD4_Naive
CD8_EMRA
CD8_EM
CD8_Eff1
CD8_Eff2
CD8_Eff3
CD8_Exh
CD8_Naive
CD8_Prolif
b
Clonotype fraction
0
0.2
0.4
0.6
0.8
1.0
0
14
28
90
Timepoint
(Days post T cell infusion)
c
UPN 394
Fraction of cells
0
0.2
0.4
0.6
0.8
1.0
Infused Product
-12
0
14
28
Timepoint
(Days post T cell infusion)
CD4_Mem
CD4_Naive
CD8_EMRA
CD8_EM
CD8_Eff1
CD8_Eff2
CD8_Eff3
CD8_Exh
CD8_Naive
CD8_Prolif
d
Clonotype fraction
0
0.2
0.4
0.6
0.8
1.0
-12
0
14
28
Timepoint
(Days post T cell infusion)
e
UPN 388
α and β chain CDR3 amino acid sequences
CVVNSPRGSYIPTF_CASSGAGRDTIYF
CALGALNDYKLSF_CASSEWGPPNTEAFF
CAASIGNFGNEKLTF_CASSPARNTEAFF
CAVKSYNTDKLIF_CSADPLFATDTQYF
CAELSGNTGKLIF_CAITPKSSYGYTF
CALEAASGQNFVF_CASSQDLSGGLFSGANVLTF
CAEIEAYNTDKLIF_CASSLDIYSNTEAFF
CAVRPEEGFGNVLHC_CASSQDGWGTDYNEQFF
CGTALYNTDKLIF_CASSTKGRLFPTEAFF
CAGQVVGGGGADGLTF_CASSPPLEGTQYF
CILRGQGAQKLVF_CASSYPGLQGYEQYF
CALSGDTGGFKTIF_CASSQSGASTDTQYF
CASPDSGNTPLVF_CASSYPGLQGYEQYF
CAATPDKLIF_CASSLGHLNQPQHF
CASYFGNEKLTF_CASKGRRWGNQPQHF
CAMSALGGQGGKLIF_CASSQDLSGGLLEDTQYF
CALSEAPEPQKLVF_CASSSVKARETQYF
CAAPYGQNFVF_CASSQDLREGLLSDTQYF
CAVSPTGFQKLVF_CASRSGASEAGELFF
CAASARSGGYQKVTF_CASSLNWGGYSIGSYNEQFF
CASPDSGNTPLVF_CASSATSGTDTQYF
CAASADTDKLIF_CASSAAGAHAGYTF
CAEKINTGFQKLVF_CASTPGIPQDTQYF
CAVDATYGGATNKLIF_CASSPDTDHNTEAFF
CALGALNDYKLSF_CSAELAGGRPAWVGANVLTF
CATEGRMDSSYKLIF_CASSHRDYGDYEQYF
CAATLYNQGGKLIF_CASSTFLEFFSDTQYF
CAVASLFDGLTF_CASSEWGPPNTEAFF
CVVTAADQFYF_CASSPNNHEQYF
CVVSEKSFGNEKLTF_CASSTFLEFFSDTQYF
CAYRSAIGDSWGKLQF_CASSLRGGNTEAFF
CATYTGTASKLTF_CASSVGGGPQETQYF
CALGALNDYKLSF_CASSQDLSGGLLEDTQYF
CASYFGNEKLTF_CASSYTGSPEQFF
CAVDRRFGNEKLTF_CSAELAGGRPAWVGANVLTF
CLGAPSGGSYIPTF_CASSYSNNLNYGYTF
CILRGQGAQKLVF_CSAELAGGRPAWVGANVLTF
CVVNRAGSSHTSGTYKYIF_CASSQDLSGGLLEDTQYF
CAYTHLNFGNEKLTF_CASSYPGLQGYEQYF
CALGALNDYKLSF_CASSYPGLQGYEQYF
f
UPN 375
α and β chain CDR3 amino acid sequences
CAMDQGGKLIF_CASSEGLGQLNQPQHF
CALSSRDKIIF_CASSPQGLTLNTEAFF
CALFSGGYQKVTF_CASSLGHRDRNYEQYF
CAMIPYSSYKLIF_CASSPSTGFLGYTF
CALLPTQGGSEKLVF_CASSIQREETQYF
CLVGSLVLNTNAGKSTF_CASKHPQGAPPYEQYF
CALLNQAGTALIF_CASSQGLEGYGYTF
CAVRDNYGQNFVF_CASSEGGTTNNEQFF
CAVTGELSNFGNEKLTF_CASSELREHPEQFF
CAVRDKGNDYKLSF_CASSQGAPLNYGYTF
CAVNHPGGYQKVTF_CASSDLAGRRYEQYF
CAMSVWDFGNEKLTF_CASSFPGLAGGRNTGELFF
CAMSAGTNAGKSTF_CASSLTGGDQPQHF
CAALSRGQNFVF_CASSLYPSRVTSGQFF
CAVSDWLTGANNLFF_CASSSGAATNYGYTF
CAVVIGGYNKLIF_CSAPRPGAPEAFF
CALDSNYQLIW_CASRNQGTYEQYF
CAAISTGTASKLTF_CASSPGTTAKNIQYF
CILRSNTGNQFYF_CASSFGKSSNQPQHF
CAMRQNTGGFKTIF_CASSRAGTNTEAFF
CAATLTGGGNKLTF_CASSPQSLEQYF
CAMREREYGNKLVF_CASGEGPNQPQHF
CVVNIQSSASKIIF_CASSRGQGATNEKLFF
CAMSAKTALIF_CASSLTRGGDTEAFF
CAMREDVGGDYKLSF_CASSLAYPTSEQFF
CALSEAGRDDKIIF_CSASVGRGYGYTF
CAMRGVNTNAGKSTF_CASSASSTRINEQYF
CAYRSALYGGSQGNLIF_CASSQGTQAYNEQFF
CATDTYGQNFVF_CASSGQTGYYEQYF
CAVTGGSYIPTF_CASSEWSLGLAGGGGNEQFF
CAMNSGGFKTIF_CASSERGGANTGELFF
CAMREVNSNSGYALNF_CASSPGQEAGANVLTF
CAVDSWGKLQF_CASSLTRGGDTEAFF
CALRGGGAGSYQLTF_CASSLSAHLPEAFF
CAEWWASKIIF_CASQAEESSYNEQFF
CAVGLGGKSTF_CASSLDPPGQGGVGNTIYF
CASNTGNQFYF_CASSPQTGTGSYGYTF
CGLQINNAGNMLTF_CASSPGSEHYEQYF
CAYRSAEAGGSQGNLIF_CASSPGVQSYEKLFF
CALDNFNKFYF_CASSLNIAGNNEQFF
g
UPN 394
α and β chain CDR3 amino acid sequences
CAGQGGNNNDMRF_CSARDRDRGFTGELFF
CALSTFQGGSEKLVF_CASSLMAGGTGELFF
CALSGPNTGNQFYF_CASTLAWGQSYNEQFF
CALSEAGGTSYGKLTF_CASSFYPGRNTEAFF
CALSEPGTGNQFYF_CASSLGRGYETQYF
CALSEAGGTSYGKLTF_CASSTPRGGVSTEAFF
CVSRDSGAGSYQLTF_CASSLTWGTGGGEQFF
CAGPNTDKLIF_CASSLTWGTGGGEQFF
CAVRDRLGADGLTF_CASSFSRLAGRFGELFF
CAVGGRGKLIF_CASSPTGVSGEKLFF
CAAGSGKLIF_CASSQVSGLDSEQYF
CAESSHYMDTGRRALTF_CASSPTSGSSYNEQFF
CLAPPQGGSEKLVF_CATSDLGTTEQFF
CAYRSVIQGAQKLVF_CSVGTGSGNEKLFF
CAVPIMNTGNQFYF_CSARVPGLAGAIVGGELFF
CAASSTFSDGQKLLF_CSARGPWVGSAFF
CAVLIYNQGGKLIF_CASSPLGGKEKLFF
CAVGGSNYKLTF_CASSFALGGDTQYF
CAAIESSGYSTLTF_CASSPGRGNEQYF
CAYRSVIQGAQKLVF_CASSGQAATNTGELFF
CAGPYSGAGSYQLTF_CASSPTGVSGEKLFF
CILRPSSGNTPLVF_CASSLIEAASYEQYF
CAALGSGGGADGLTF_CASRYRSDTQYF
CALSGSARQLTF_CSATFSGRVDEQFF
CAVRDKANTGFQKLVF_CASSIGAGDLSGEQFF
CALSEGGFQGAQKLVF_CASSLTSGGALHGLNEQFF
CAASGEDDKIIF_CASSSIGGGQGYTF
CAVKGGNQFYF_CASSQGLPGELFF
CAESDNQGGKLIF_CASSYSGEDTQYF
CAVQSGYSTLTF_CASREDTSGTEAFF
CASQPRDSNYQLIW_CASSSIGGGQGYTF
CALCADRGSTLGRLYF_CASSFLPGASTDTQYF
CAYRGSNAGGTSYGKLTF_CSARDRDRGFTGELFF
CAVRDRLGADGLTF_CASSHGQGSYEQYF
CAASKPGGYNKLIF_CASSHGQGSYEQYF
CAASVKTSYDKVIF_CASSFALGGDTQYF
CAVRDRLGADGLTF_CASSLWQGQGANNEQFF
CAMRGHNTNAGKSTF_CASSGQAATNTGELFF
CAGDQGAQKLVF_CASSHGQGSYEQYF
CAVRSYNTDKLIF_CASSLEGQGSEAFF

Figure 5.4 Single cell analysis of CD3+ T cell subsets and TCRα/β repertoire diversity in the peripheral blood. **(a)** scRNAseq analysis of CD3+ T cell subsets in the infused product and in the peripheral blood T cells at indicated timepoints post-T cell infusions of UPN375. **(b)** Single cell analysis of TCRα/β repertoire diversity in the peripheral blood T cells of UPN375 at indicated timepoint post-T cell infusion. **(c-d)** Same as (a-b) for UPN394. **(e)** Legend for Figure 5.3g. **(f)** Legend for Figure 5.4b. **(g)** Legend for Figure 5.4d.

Immune landscape changes in patient with biochemical response

Endogenous and CAR T cell populations in peripheral blood were further characterized by flow cytometry, as well as by single cell RNA sequencing (scRNAseq) and TCR repertoire analysis in this patient. Initial assessment of T cell subsets in peripheral blood pre- and post- LD and CAR T cell infusion in this patient showed dynamic changes in naïve (Tn), central memory Tcm), effector memory (Tem), and terminally differentiated effector memory (Temra) cells over time. Greater re-emergence of CD8+ Tcm and Tem cells were observed by day 28 post-CAR T cell infusion. CD8+ CAR T cells expanded with this phenotype by day 14 in this patient. Interestingly, CAR+ and endogenous non-CAR T cells showed increased PD1 expression (and smaller increases in LAG3 and TIM3) over the 28 days following treatment, which is associated with an activation and/or exhaustive phenotype. Peripheral blood CAR T cells showed elevated expression of CX3CR1, which has been correlated with response to immunotherapy with anti-PD1 immune checkpoint blockade (10). Few CX3CR1-positive T cells were observed in the product prior to infusion. Similar results in CAR+ and endogenous non-CAR T cells in the peripheral blood were observed in a patient in DL3, but not in DL1. scRNAseq corroborated these data, with increased effector CD8+ T cell subsets including CX3CR1+ CD8+ T cells in patients (Figure 5.3f and Figure 5.4a, c). Single cell TCR a/b repertoire analysis of endogenous T cells in peripheral blood demonstrated emerging and expanded clones by day 28 post-CAR T cell infusion in patients (Figure 5.3g, Figure 5.4b, d, e-g, and Figure 5.5), which contracted at days 90 in UPN388, suggesting TCR clonal diversity changes following therapy. Collectively, these data suggest that LD + PSCA-CAR T cell therapy can induce biochemical and radiographic response along with changes in the immune landscape and TCR repertoire.

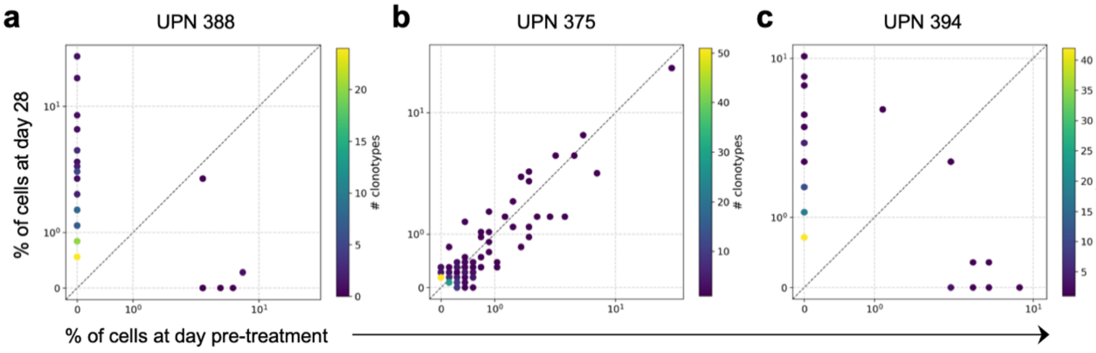


Figure 5.5 TCR repertoire diversity in peripheral blood T cells. Number of cells per clonotype at day 0 versus day 28 following therapy in UPN388 **(a)**, UPN375 **(b)**, and UPN394 **(c)**.

Discussion

CAR T cell therapy has achieved durable response rates for patients with refractory hematological malignancies (16-18), creating enthusiasm in translating this therapy to patients with solid tumors. Our study evaluated PSCA-directed CAR T cells in patients with mCRPC. We observed biochemical and radiographic responses in patients following LD and PSCA-CAR T cell infusion. The dose-limiting toxicity was cystitis, which was likely an on-target/ off-tumor effect (19) with contribution from cyclophosphamide LD (20). Reducing the cyclophosphamide dose avoided high grade cystitis events in DL3 while retaining similar peripheral blood expansion of CAR T cells, although small number of patients limits the statistical power to exclude a difference. In this heavily pretreated population, encouraging anti-cancer responses were seen. Our findings are limited by the small number of subjects accrued. Accrual to a phase 1 trial with tissue pre-screening requirement, holds to accrual during DLT assessment periods and enrollment of heavily pre-treated patients was by nature slow, and many patients did not proceed with treatment if disease progression occurred in a way that led to ineligibility, including some who had undergone leukapheresis. The relatively lengthy process may have excluded patients with more aggressive disease or borderline performance status. This highlights the importance of streamlining enrollment in phase 2 to reduce enrollment bias and overall study cost by improving the rate of infusion of manufactured CAR T cell product. This study validates PSCA as a viable CAR T cell therapeutic target and provides encouraging early clinical data to support further studies, focused on extending CAR T cell persistence, which, with the use of novel dosing and/or combinatorial strategies is hoped to lead to improved responses in patients.

Both activity and toxicity of CAR T were impacted by the addition of LD, though the role of LD in facilitating CAR T cell activity in solid tumors is likely different than the role it plays in hematological malignancies. Preconditioning with LD promoted greater peripheral blood CAR T cell expansion and serum cytokine levels which manifested in greater objective anti-cancer response in DL2 and DL3. These phase 1 trial results validate our recent preclinical studies, which found that increased efficacy of CAR T cells following administration of cyclophosphamide was associated with enhanced T cell infiltration into tumors along with antigen presenting cell (i.e., dendritic cell) infiltration and reduced myeloid suppressive features compared to CAR T cell therapy alone (21). Notably, lower dose cyclophosphamide (300 mg/m2) still yielded greater CAR T cell bioactivity than absence of LD, while it did reduce the toxicities compared to cyclophosphamide dosed at 500 mg/m2; similar findings have been documented in hematological malignancies (22). Given the critical role of LD, an important avenue of investigation will be to study different LD regimens for optimal changes in the tumor immune microenvironment. Preclinically, gene ontology enrichment analysis identified T cell migration and IFNγ production as key processes enhanced by cyclophosphamide pre-treatment (21) and these can be used as endpoints of preclinical exploration. Metronomic dosing strategies of cyclophosphamide and alternative LD regimens warrant evaluation since the traditional high dose IV 3-day LD regimen adopted from hematological malignancy CAR T cell trials may not be equivalently translated for solid tumors. Taxanes and platinum agents have shown potential for modulation of the tumor immune microenvironment and for solid tumor CAR T cell therapy (23,24) and bendamustine is also emerging as a potentially valuable LD agent (22).

DLTs were only seen after the addition of LD in this trial, mirroring the toxicity experience of the PSMA-targeted TGFβ dominant negative CAR T cell trial, in which DLT were only encountered after LD chemotherapy was added (25). In the PSMA-targeting TGFβ-insensitive armored CAR T cell trial the dose of CAR T cells was reduced after high grade toxicity occurred (sepsis and macrophage activation syndrome/ hemophagocytic lymphohistiocytosis – MAS/HLH). Our approach of incorporating LD prior to CAR T cell dose escalation allowed for earlier appearance of DLTs during the trial. Therefore, reducing LD and maintaining CAR T cell dose resulted in continued evidence of anti-cancer efficacy with a lower toxicity profile. CRS onset was slightly delayed with PSCA CAR T cell therapy compared to the experience in hematological malignancies (26), with median onset at 4 days post infusion in this study. Tocilizumab was administered in 3 patients primarily for relief of fever and chills, with no grade 3 CRS events, and no hypotension or hypoxia events noted. Unlike other CAR T cell trials in mCRPC (25, 27,28), no high-grade neurologic toxicity nor MAS/HLH events occurred though it is unclear whether the PSCA target or this particular CAR cell construct underly this observation. Overall, the favorable toxicity profile of PSCA-CAR T cell therapy enables the currently accruing phase 1b trial (NCT05805371) to proceed with entirely outpatient dosing in the context of close clinical monitoring.

While PSCA CAR T expansion was robust with objective measures of disease-modifying activity including PSA decline, reduction in CTCs, and radiographic improvements, there was a lack of CAR T cell persistence that corresponded to the lack of durable remission. Innovative strategies will be needed to enhance CAR T cell persistence and prolong the anti-cancer efficacy. Armoring CAR T cells with modifications such as the TGFb-dominant negative approach with PSMA targeting showed efficacy but at the cost of fatal toxicities, perhaps due to uncontrolled over-expansion (25). Over-expansion was also potentially the cause of fatal toxicity with the strategy employed by the Go-CART agent, in which rimiducid was administered in pulses to stimulate proliferation (28). Alternative strategies to improve persistence of CAR T cells may include enriching for T naïve/stem/memory cells (29) and incorporating agents into the CAR T cell manufacturing process to improve T cell fitness, including AKT inhibitors (30, 31). Allogeneic cell therapy approaches may further modify therapeutic activity (32) while also increasing feasibility by shortening time to treatment, which was a factor in the high drop-out rate observed in this trial. In order to avoid high-grade toxicity and to prolong the presence of infused T cells, our phase 1b strategy will administer multiple smaller doses of PSCA-CAR T cells rather than escalating to a larger single dose. Preclinical models of cystitis can be leveraged to study potential prevention or early intervention strategies, which could make it feasible to escalate the PSCA-CAR T cell dose in future trial iterations.

Tumor antigen heterogeneity with neuroendocrine transformation was noted in one patient in this study and may be a more general resistance mechanism in heavily-treated mCRPC patients. Treatment-emergent neuroendocrine transformation is reported to occur more commonly in mCRPC recently since the introduction of powerful androgen receptor pathway inhibitors (33) Thus, treating patients with mCRPC earlier in the disease course may be necessary to achieve durable responses. Alternatively, dual targeting to include proteins expressed on the de-differentiated CRPC cell populations such as CEA or DLL3 (34) may be required. T cell exhaustion may have contributed to the limited duration of activity, as indicated by upregulation of PD1 in peripheral CAR T cells. Anecdotal experiences suggest that CAR T cells lose function in the setting of high tumor volumes, and immune checkpoint inhibitor therapy may rescue incomplete CAR T cell responses (35) which may have contributed to the long-term survival of a patient who received pembrolizumab as part of a clinical trial after participation in the phase 1 PSCA-CAR T study (Figure 5.2).

In summary, our first-in-human phase 1 trial evaluating PSCA-CAR T cell therapy showed bioactivity and early evidence of clinical effectiveness, though on-target toxicity of cystitis impacted intended CAR T cell dose escalation. Reduced LD dose mitigated toxicity while still enhancing CAR T cell expansion compared to no LD. Future studies will explore multi-dose and combinatorial strategies to improve persistence with the goal of increasing clinical activity in patients with mCRPC.

Methods

Trial design and patients

This was a single-center phase 1 trial aimed at evaluating safety and feasibility of intravenously administered, lentivirally transduced PSCA-CAR T cells in patients with mCRPC, with a total of three dose level (DL) cohorts. The primary endpoints were safety and dose-limiting toxicities (DLT). The secondary endpoints were persistence of CAR T cells to 28 days post infusion (defined as CAR T cells comprising at least 7.5 copies/µg of DNA of total CD3 cells), expansion of CAR T cells (Max log10 copies/µg of genomic DNA, disease response (PSA decline, RECIST) and survival described as percent of participants alive at 6 months. Exploratory endpoints were phenotypes and frequencies of immune cell subsets in the peripheral blood pre- and post-therapy, phenotype of tumor-infiltrating lymphocytes, gene expression (by RNA-seq) of CTCs, cfDNA in peripheral blood by whole exome sequencing, CAR immunogenicity (anti-PSCA-CAR antibodies) and serum cytokine profile before and after CAR T cell infusion to assess potential CRS toxicity and CAR T cell effector function.

The trial was conducted in accordance with the United States Food and Drug Administration (FDA) and International Conference on Harmonization Guidelines for Good Clinical Practice, the Declaration of Helsinki and applicable institutional review board requirements (study protocol approved by the City of Hope Institutional Review Board). Only subjects with male sex were enrolled due to prostate cancer presenting only in this sex group. After IND was obtained and institutional review board approved the protocol, subjects provided written informed consent in a two-step process. Many patients pre-screened (tissue PSCA testing) but did not proceed with leukapheresis due to lengthy wait times with limited slot availability and accrual pauses during DLT evaluation periods. The trial was registered with clinicaltrials.gov (NCT03873805). The City of Hope Data

Safety Monitoring Board monitored the conduct of this study to ensure the safety of enrolled and treated subjects, and the validity and integrity of the acquired data.

A starting dose of 100 million (M) PSCA-CAR T cells was selected based on experience with other CAR T cell trials and anticipated effective dose. The first 3 subjects on Dose level (DL) 1 at 100M CAR T cells without fludarabine and cyclophosphamide (Flu/Cy) preconditioning lymphodepletion (LD), and the first 3 subjects on DL2 at 100M CAR T cells with LD were staggered through the dose-limiting toxicity (DLT) period. All further subjects were accrued to dose levels (DLs) in cohorts of 3. After evaluation of the data from the completed DLT period (28 days) the protocol management team met to determine whether it was safe to escalate to the next DL, with rules following the TEQR design of Blanchard and Longmate (11) with an equivalence range of 0.20-0.35 and a too toxic level of 0.51. The first cohort received 100M CAR T cells without LD; the subsequent cohorts would all receive LD with plans to escalate the dose of CAR T cells from 100M to 300M to 600M, and the option to de-escalate the dose to 50M if LD plus 100M CAR T cells was not tolerated.

Lymphodepletion (LD) chemotherapy: standard regimen of cyclophosphamide 500 mg/m2 IV on days -5 to -3 and fludarabine 30 mg/m2 IV on days -5 to -3 was employed in DL2; this was reduced due to DLT, and DL3 subjects received cyclophosphamide 300 mg/m2 IV on days -5 to -3 with the same dose schedule of fludarabine. Prophylactic G-CSF was not utilized, but G-CSF could be added for neutropenia if the treating physician felt it was indicated, as well as all other standard supportive measures such as antiemetics.

In order to attempt to exclude patients unlikely to benefit due to lack of tumor PSCA expression, all potential subjects signed a pre-screening consent form so that archived tissue could be tested for PSCA by immunohistochemistry (IHC) staining. Subjects were required to have at least moderate PSCA expression in their prostate primary or metastatic biopsy tissue to enroll in the study, although due to lack of a validated assay there was no pre-specified cut-off. All subjects enrolled had PSCA expression in >30% of tumor cells (Figure 5.1a CONSORT diagram). An on-study biopsy was performed, and for soft tissue metastases confirmation of PSCA staining was required (this was not required for bone metastases due to inadequate calibration of the IHC assay on bone material); repeat biopsy of the same metastatic area was performed during the day 28 assessment period.

Patients with mCRPC were eligible if they had experienced disease progression on at least one androgen receptor pathway inhibitor, e.g. abiraterone, enzalutamide. Prior taxane chemotherapy was allowed but not required. Creatinine clearance > 50 mL/min were required, as well as AST/ALT < 5 x ULN and bilirubin < 2.0 mg/dL. Electrocardiogram was required to show no acute abnormalities requiring intervention and echocardiogram was required to document a left ventricular ejection fraction of > 40%. Patients with clinically significant cardiac arrhythmias or central nervous system disease were excluded. Patients with HIV, active hepatitis B or C, or uncontrolled active infection were excluded. Eligibility was confirmed prior to leukapheresis and again prior to start of treatment (DL1: CAR T and DL 2 or DL 3: LD).

CAR T cell manufacturing

Following screening and enrollment into the trial, subjects underwent leukapheresis at City of Hope's Michael Amini Transfusion Medicine Center. Autologous PBMC were immunomagnetically depleted of CD14+ and CD25+ cells, then stimulated with CD3/CD28 DynaBeads and subjected to transduction with PSCA(dCH2)BBζ/CD19t lentivirus (multiplicity of infection = 0.1) followed by T cell expansion for 10-16 days until the freezing process. Cells were manufactured in the City of Hope Center for Biomedicine and Genetics (CBG) GMP facility; details are provided in the clinical protocol (see Supplemental Materials).

Flow cytometry

Peripheral blood samples were obtained from subjects prior to and at various timepoints for 28 days following CAR T cell infusion, as well as day 60, 90, and q12 weeks after day 90 to evaluate CAR T cell expansion/persistence. Peripheral blood samples were lysed using BD PharmLyse (15 min at RT) and quenched using RPMI containing 10% FBS. Cells were resuspended in FACS buffer (Hank's balanced salt solution without Ca2+, Mg2+, or phenol red (HBSS−/−, Life Technologies) containing 2% FBS and 1 × AA). Cells were incubated with Fc block (BD Biosciences) for 5 min at RT and then incubated with fluorescence-labelled antibodies for 15 min at RT in the dark. Unless otherwise stated, antibodies were used at a dilution of 1:100. Cell viability was determined using 4′, 6-diamidino-2-phenylindole (DAPI, Sigma, Cat: D8417). For samples run on the Cytek Aurora, samples were thawed, counted using a Muse cell counter (1 milion cells) and were stained in a two-step process. Before staining, the cells were Fc Blocked with BD Pharmingen™ Human BD Fc Block™ (BD Biosciences) for 20 min on ice, washed, spun, and resuspended in the first master mix. The first master mix included 1 antibody, PDL1 PE-Fire810 (Biolegend), in FACS buffer. Following incubation on ice for 20 minutes, cells were washed twice with FACS Buffer and then stained with a 23-antibody master mix. The second master mix was prepared using FACS Buffer with Brilliant Buffer Plus (BD Horizon). After incubation, the cells were washed twice with FACS Buffer and finally resuspended in FACS buffer with 7-AAD (Invitrogen). Flow cytometry was performed on a MACSQuant Analyzer 10 (Miltenyi Biotec) or Cytek Aurora 3, and data were analyzed with FlowJo software (v10.8.1, TreeStar) or OMIQ software (Dotmatics).

Single cell transcriptomics and TCR repertoire analysis

Single cell RNA and TCR libraries were prepared using 10x Genomics Chromium Single Cell Immune Profiling Solution Kit and workflow (10×Genomics Inc.). Cells were thawed, washed twice, and resuspended in RPMI containing 10% FBS to a final concentration of 100–1000 cells per µl as determined by Cell Countess. Samples with unique donor identities were pooled together and processed for a targeted cell recovery of 10,000 cells. Single cell RNA-seq and TCR-seq libraries were assessed for quality and quantified using the Agilent 2100 Bioanalyzer System and Qubit 3.0 Fluorometer. Single cell RNA libraries were sequenced on an Illumina NovaSeq to a minimum sequencing depth of 25,000 reads per cell using read lengths of 26 bp read 1, 8 bp i7 index, 98 bp read 2. The single-cell TCR libraries were sequenced on an Illumina HiSeq and NovaSeq to a minimum sequencing depth of 5,000 reads per cell using read lengths of 150 bp read 1, 8 bp i7 index, 150 bp read 2. DNA was extracted from each sample donor's CAR T cell product using the DNeasy Blood and Tissue Kit (Qiagen) and recommended protocol for Purification of Total DNA from Animal Blood or cells. Isolated DNA was genotyped with Infinium Omni5-4 Beadchip Array at City of Hope's Integrative Genomics Core.

A full description of the methods and code used to process and analyse the single-cell RNA seq data is available at:
https://github.com/pachterlab/DBALLSMRDMCMGWSTPMBDKPFP_2023.

Acknowledgements

This study was funded by a 2017 Prostate Cancer Foundation Challenge Award (S.J.P., S.J.F.) and an NCI SPORE Award (P50CA092131). Studies reported in this publication were also supported by the Doug and Rhonda Collier Foundation fund, the Fiterman Family Foundation fund, and the Borstein Foundation fund. Work performed in the GMP facility and the COH Pathology Core supported by the National Cancer Institute of the National Institutes of Health under grant number P30CA033572. We thank the Analytical Pharmacology Core Facility (APCF) for their assistance in performing correlatives assays, and the T Cell Therapeutics Research Laboratories (TCTRL) and the Center for Biomedicine and Genetics (CBG) for CAR T cell manufacturing and product release. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes.

Competing Interests

T.B.D. is a consultant for AstraZeneca and Janssen. S.J.P. and S.J.F. are scientific advisors to and receive royalties from Mustang Bio. S.J.P. is also a scientific advisor and/or receives royalties from Imugene Ltd, Adicet Bio, Port Therapeutics, and Celularity. S.J.P. and S.J.F. are listed as co-inventors on a patent on chimeric antigen receptors targeted to PSCA, which is owned by the City of Hope. All other authors declare that they have no competing interests.

Data availability statement

All required clinical data have been uploaded to clinicaltrials.gov. All requests for raw and analyzed data and materials should be addressed to the corresponding authors and will be reviewed by the institution to verify whether the request is subject to any intellectual property or confidentiality obligations. Patient data may be subject to patient confidentiality. Any data and materials that can be shared will be released via a material transfer agreement.

Code availability statement

A description of the methods and the code used to process and analyze the single-cell RNA seq and TCR seq data is available at:
https://github.com/pachterlab/DBALLSMRDMCMGWSTPMBDKPFP_2023.
Full access the TCR and single cell RNA seq can be accessed via:
https://www.dropbox.com/scl/fo/rhgr2y28az1e2h0avaj0l/h?rlkey=6a4wlnus0png19mb80g42uja1&dl=0 (password: coh_128781)

Obtaining biologic materials statement

Chimeric antigen receptor T cells (CAR T cells) were manufactured at City of Hope in the GMP facility, with materials and processes approved by FDA IND. These are provided (administered) only to individual patients enrolled on the trial.

References

1. Siegel RL, Miller KD, Fuchs HE, Jemal A. Cancer Statistics, 2021. *CA Cancer J Clin* 2021; 71:7-33.

2. Higano CS, Corman JM, Smith DC, Centeno AS, Steidle CP, Gittleman M, *et al.* Phase 1-2 dose-escalation study of a GM-CSF-secreting, allogeneic, cellular immunotherapy for metastatic hormone-refractory prostate cancer. *Cancer* 2008; 113:975-84.

3. Gulley JL, Borre M, Vogelzang NJ, Ng S, Agarwal N, Parker CC, et al. Phase III trial of PROSTVAC in asymptomatic or minimally symptomatic metastatic castration-resistant prostate cancer. *J Clin Oncol* 2019; 37:1051-61.

4. Antonarakis ES, Piulats JM, Gross-Goupil M, Goh J, Ojamaa K, Hoimes CJ*, et al.* Pembrolizumab for Treatment-Refractory Metastatic Castration-Resistant Prostate Cancer: Multicohort, Open-Label Phase II KEYNOTE-199 Study. *J Clin Oncol* 2020;38(5):395-405 doi 10.1200/JCO.19.01638.

5. Kwon ED, Drake CG, Scher HI, Fizazi K, Bossi A, van den Eertwegh AJ*, et al.* Ipilimumab versus placebo after radiotherapy in patients with metastatic castration-resistant prostate cancer that had progressed after docetaxel chemotherapy (CA184-043): a multicentre, randomised, double-blind, phase 3 trial. *Lancet Oncol* 2014;15(7):700-12 doi 10.1016/S1470-2045(14)70189-5

6. Kantoff PW, Higano CS, Shore ND, Berger ER, Small EJ, Penson DF*, et al.* Sipuleucel-T immunotherapy for castration-resistant prostate cancer. *N Engl J Med* 2010;363(5):411-22 doi 10.1056/NEJMoa1001294

7. Dorff TB, Narayan V, Forman SJ, Zang PD, Fraietta JA, June CH, Haas NB, Priceman SJ. Novel redirected T-cell immunotherapies for advanced prostate cancer. *Clin Cancer Res* 2022; 28:576-84.

8. Gu Z, Thomas G, Yamashiro J, Shintaku IP, Dorey F, Raitano A, *et al.* Prostate stem cell antigen (PSCA) expression increases with high gleason score, advanced stage and bone metastasis in prostate cancer. *Oncogene* 2000; 19:1288-96.

9. Priceman SJ, Gerdts EA, Tilakawardane D, Kennewick KT, Murad JP, *et al.* Co-stimulatory signaling determines tumor antigen sensitivity and persistence of CAR T cells targeting PSCA + metastatic prostate cancer. *Oncoimmunology* 2017; 7:e1380764

10. Yamauchi T, Hoki T, Oba T, Jain V, Chen H, Attwood K, *et al.* T cell CX3CR1 expression as a dynamic blood-based biomarker of response to immune checkpoint inhibitors. *Nat Commun* 2021; 12:1402 doi: 10.1038/s41467-021-21619-0.

11. Blanchard MS, Longmate JA. Toxicity equivalence range design (TEQR): a practical phase I design. *Contemp Clin Trials* 2011; 32:114-21.

12. Chai S, Matsumoto N, Storgard R, Peng C-C, Aparicio A, Ormseth B, *et al.* Platelet-coated circulating tumor cells are a predictive biomarker in patinets with metastatic castrate-resistant prostate cancer. *Mol Cancer Res* 2021; 19:2036-45

13. Shishido SN, Sayeed S, Courcobubetis G, Djaladat H, Miranda G, Pienta KJ, *et al.* Characterization of cellular and acellular analytes from pre-cystectomy liquid biopsies in patients newly diagnosed with primary bladder cancer. *Cancers* 2022; 14:758. Doi: 10.3390/cancers14030758

14. Setayesh SM, Hart O, Naghdloo A, Hga N, Nieva J, Lu J, Hwang S, *et al.* Multianalyte liquid biopsy to aid the diagnostic workup of breast cancer. *NPJ Breast Cancer* 2022; 8:112. Doi:10.1038/s41523-022-00480-4.

15. Shishido SN, Ghoreifi A, Sayeed S, Courcobetis G, Huang A, Ye B, *et al.* Liquid biopsy landscape in patinets with primary upper tract urothelial carcinoma. *Cancers* 2022; 14:3007. Doi:10.3390/cancers14123007.

16. Neelapu SS, Locke FL, Bartlett NL, Lekakis LJ, Miklos DB, *et al.* Axicabtagene ciloleucel CAR T-cell therapy in refractory large B-cell lymphoma. *N. Engl. J. Med.* 2017;**377**:2531–2544

17. Maude SL, Laetsch TW, Buechner J, Rives S, Boyer M, *et al.* Tisagenlecleucel in children and young adults with B-cell lymphoblastic leukemia. *N. Engl. J. Med.* 2018;**378**:439–448

18. Schuster SJ, Svoboda J, Chong EA, Nasta SD, Mato AR, *et al.* Chimeric antigen receptor T cells in refractory B-cell lymphomas. *N. Engl. J. Med.* 2017;**377**:2545–2554

19. Cheng L, Reiter RE, Jin Y, Sharon H, Wieder J, Lane TF*, et al.* Immunocytochemical analysis of prostate stem cell antigen as adjunct marker for detection of urothelial transitional cell carcinoma in voided urine specimens. *J Urol* 2003;**169**(6):2094-100 doi 10.1097/01.ju.0000064929.43602.17.

20. Almalag HM, Alasmari SS, Alrayes MH, Binhameed MA, Alsudairi RA, *et al.* Incidence of hemorrhagic cystitis after cyclophosphamide therapy with or without mesna: a cohort study and comprehensive literature review. *J Oncol Pharm Pract* 2021; 27:340-9. Doi: 10.1177/1078155220920690. city

21. Murad JP, Tilakawardane D, Park AK, Lopez LS, Young CA, Gibosn J *et al.* Pre-conditioning modifies the TME t enhance solid tumor CAR T cell efficacy and endogenous protective immunity. *Mol ther* 2021; 29:2335-49.

22. Amini L, Silbert SK, Maude SL, Nastoupil LJ, Ramos CA, *et al.* Preparing for CAR T cell therapy: patient selection, bridging therapies and lymphodepletion. *Nat Rev Clin Oncol* 2022; 19:342-55.

23. Alzubi J, Dettmer-Monaco V, Kuehle J, Thorausch N, Seidl M, Taromi S*, et al.* PSMA-Directed CAR T Cells Combined with Low-Dose Docetaxel Treatment Induce Tumor Regression in a Prostate Cancer Xenograft Model. *Mol Ther Oncolytics* 2020;**18**:226-35 doi 10.1016/j.omto.2020.06.014

24. Kershaw MH, Devaud C, John LB, Westwood JA, Darcy PK. Enhancing immunotherapy using chemotherapy and radiation to modify the tumor microenvironment. *Oncoimmunology* 2012; e25962.

25. Narayan V, Barber-Rtenberg JS, Jung I-Y, Lacey SF, Rech AJ, *et al.* PSMA-targeting TGFb-insensitive armored CAR T cells in metastatic castration-resistant prostate cancer: a phase 1 trial. *Nat Med* 2022; 28:7224-34.

26. Maude, SL, *et al.,* Managing cytokine release syndrome associated with novel T cell-engaging therapies. *Cancer* J, 2014. **20**(2): p. 119-22

27. Slovin SF, Dorff TB, Falchook GS, Wei XX, Gao X, McKay RR, *et al.* Phase 1 study of P-PSMA-101 CAR-T cells in patients with metastatic castration-resistant prostate cancer. *J Clin Oncol* 2022; supp (abstr 98).

28. Stein MN, Teply BA, Gergis U, Strickland D, Senesac J, Bayle H, *et al.* Early results form a phase 1, multicenter trial of PSCA-specific GoCAR T cells (BPX-601) in patients with metastatic castration-resistant prostate cancer (mCRPC). *J Clin Oncol* 2023; 41 (supp) abstr 140

29. Aldoss I, Khaled SK, Wang X, Palmer J, Wang Y, Wagner JR, Clark MC, *et al.* Favorable activity and safety profile of memory-enriched CD19-targeted chimeric antigen receptor T-cell therapy in adults with high-risk relapsed/refractory ALL. *Clin Cancer Res* 2023; 29:742-53.

30. Urak R, walter M, Lim L< Wong CLW, Budde LE, Thomas S, Forman SJ, Wang X. Ex vivo Akt inhibitiorn promotes the generation of potent CD19 CAR T cells for adoptime immunotherapy. J IMmunother Cancer 2017; 5:26 doi:10.1186/s40425-017-0227-4

31. Mehra V, Agliardi G, Pinto JDA, Shafat MS, Garai AC, Green L, *et al.* AKT inhibition generates potent polyfunctional clinical grade AUTO1 CAR T-cells, enhancing function and survival. *J Immunother Cancer* 2023; 11:e007002. Doi:10.1136/jitc-2023-007002.

32. Depil S, Duchateau P, Grupp SA, Mufti G, Poirot L. ‘Off the shelf; allogeneic CAR T cells: development and challenges. *Nat Rev Drug Discov* 2020; 19:185-99.

33. Liu S, Alabi BR, Yin Q, Stoyanova T. Molecular mechanisms underlying the development of neuroendocrine prostate cancer. Semin Cancer Biol 2022; 86:57-68.

34. DeLucia DC, Cardillo TM, Ang L, Labrecque MP, Zhang A, Hopkins JE *et al.* Regulation of CEACAM5 and therapeutic efficacy of an anti-CEACAM5-SN38 antibody-drug conjugate in neuroendocrine prostate cancer. *Clin Cancer Res* 2021; 27:759-774

35. Adusumilli PS, Zauderer MG, Riviere I, Solomon SB, Rusch VW, O’Cearbhaill RE, *et al.* A phase I trial of regional mesothelin-targeted CAR T-cell therapy in patients with malignant pleural disease, in combination with the anti-PD-1 antibody agent pembrolizumab. *Cancer Discovery* 2021; 11:2748-63.

TRANSCRIPTOMICS IN HEALTHCARE – PART II

Analysis of Heterogenous Datasets Across Patients: Dealing with Varying Medical Histories, Sequencing Technologies, and Treatment Time Points

Preamble

While transcriptomics technologies have revolutionized the study of RNA expression in healthcare, the heterogeneity of data obtained from different patients with varying medical histories, in addition to the technical and biological noise inherent to sequencing data, results in substantial and unique data analysis challenges. The data presented in the previous subchapter was derived from 12 prostate cancer patients at varying time points of CAR T therapy from CAR T product, peripheral blood, and solid tumor tissue using two different sequencing technologies, single-cell gene expression and V(D)J immune repertoire sequencing, resulting in a total of 32 multiplexed datasets for each sequencing technology. In this subchapter, I will describe additional data analysis approaches, expanding on the methods and results described in the previous subchapter and emphasizing the consequences of data heterogeneity and complexity.

Methods and Results

Detection of transgenes from single-cell RNA sequencing data

Since the single-cell RNA sequencing data described in the previous subchapter was generated from patients who were treated with transgenic T cells, we can detect these cells in the sequencing data based on their expression of the transgenic CAR product. This can be achieved by manually adding the CAR sequence to the reference genome (excluding any endogenous sequences such as CD19). The detection rate of transgenes is low. Hence, this approach works best for samples with a high copy number of transgenes, such as the

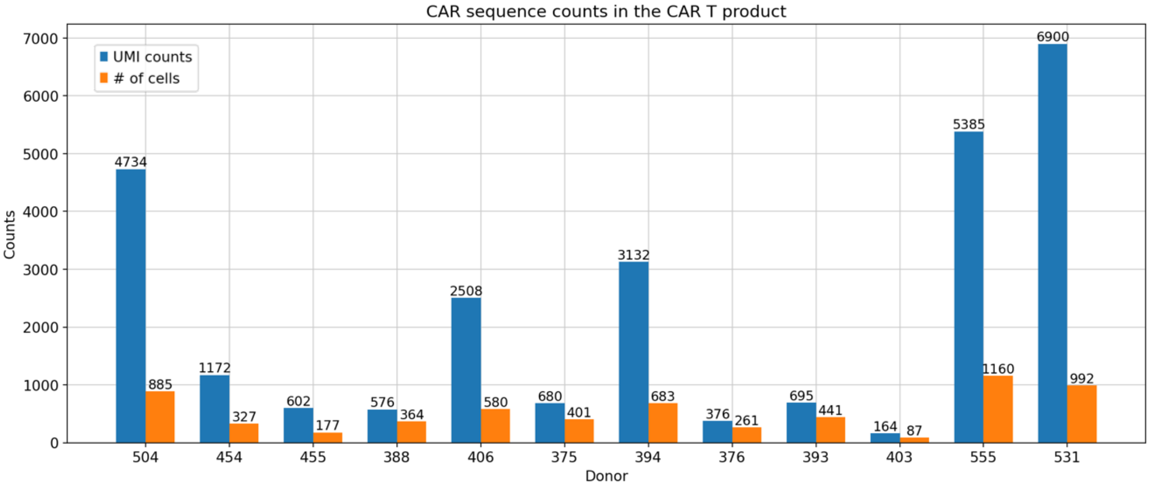


Figure 5.6 UMI counts of the CAR transgene and the number of CAR+ cells obtained for each donor.

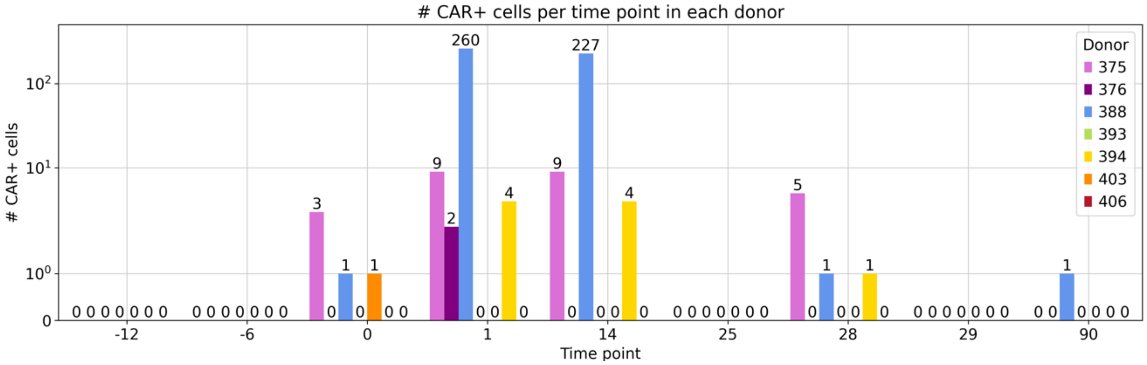


Figure 5.7 The number of transgenic CAR+ cells per donor at each treatment time point.

CAR T product (Figure 5.6). Interestingly, transgenic CAR T cells were only detected in the peripheral blood samples of the two patients that showed the most promising treatment response (Figure 5.7).

Data quality control and filtering

Following the alignment of raw sequencing data to a reference genome, the data quality needs to be assessed and low-quality cells removed. Knee and library saturation plots are often generated to evaluate data quality. A 'knee plot' shows the number of unique reads (each read tagged with a Unique Molecular Identifier (UMI)) observed for each cell and is used to determine a threshold above which cells are considered valid. Cells with relatively few reads are considered low-quality and removed before further analysis. A library saturation plot shows the number of genes detected over the total read count for each cell. The library plot plateaus as the introduction of more UMIs does not lead to the detection of new genes, thereby indicating a saturation with UMIs. Extended Data Figure 5.1 shows the knee and library saturation plots for all 32 gene expression datasets obtained from the 12 prostate cancer patients throughout different time points. Several of these datasets are multiplexed across patients. The data quality and barcode filtering threshold differ greatly between datasets (Extended Data Figure 5.1). Due to the heterogeneity in quality between datasets, it is crucial to independently assess data quality and filter cells for each dataset. Otherwise, some datasets would be filtered too conservatively, while others would have introduced low-quality cells into the analysis.

Batch correction and clustering

Following data quality assessment and filtering, the datasets were concatenated, and log(CP10k +1) normalized for further analysis. Given that datasets spanning different patients, time points, and sequencing runs were combined, we considered batch correction to remove experimental biases and batch effects prior to clustering. However, after batch correction using single-cell variational inference (scVI)[1], the gene expression profiles were incorrectly homogenized across the datasets. This was easily identified by the suddenly widespread expression of the transgenic CAR construct in cell types other than T cells. Instead, we clustered the combined dataset using Leiden clustering[2] without prior batch correction and examined whether the clustering was best explained by sequencing batches

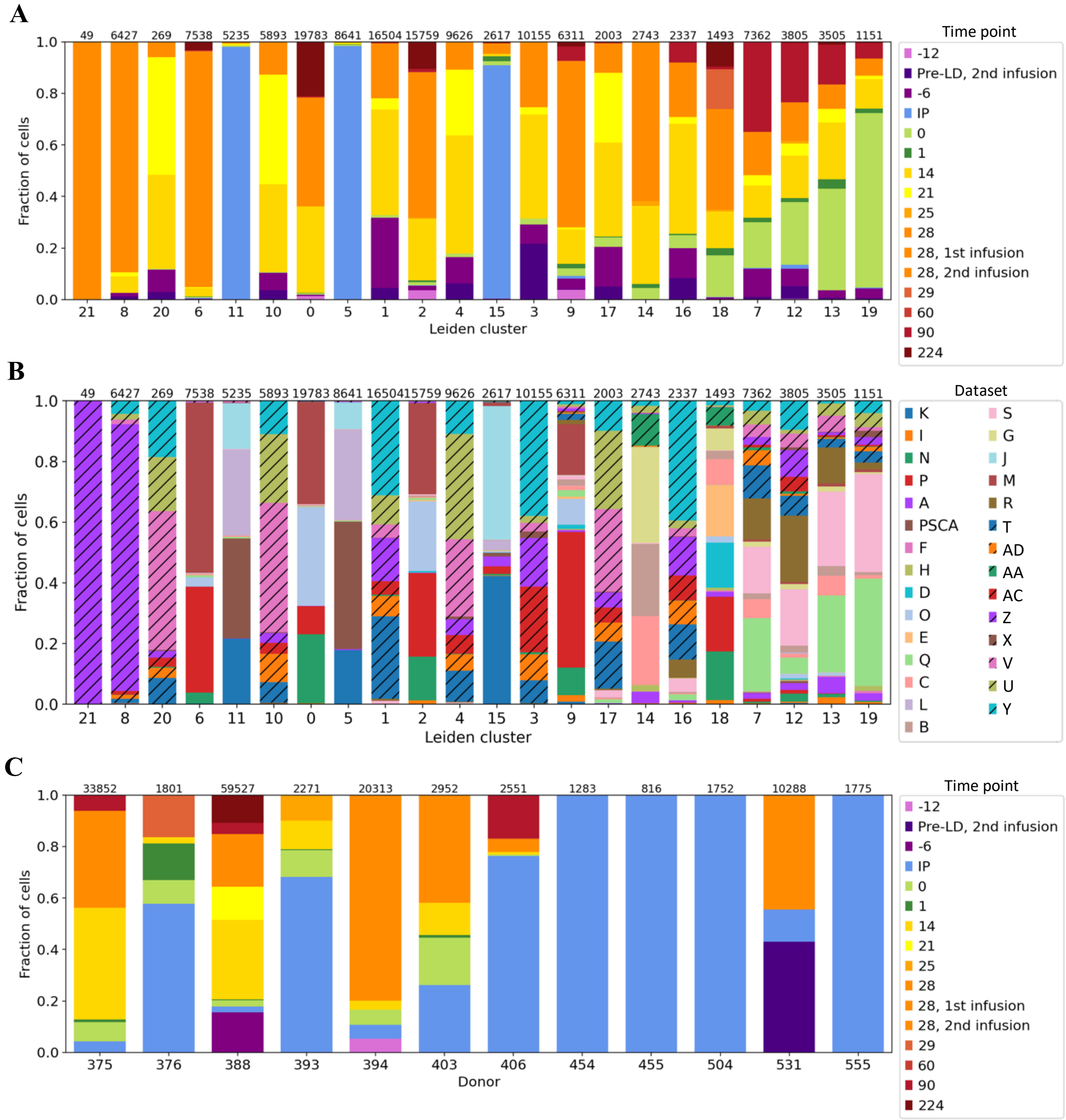


Figure 5.8 A The fraction of cells obtained for each time point per Leiden cluster. **B** The fraction of cells from each sequencing batch (dataset) in each Leiden cluster. **C** The fraction of cells obtained for each time point per donor.

or the expected biological groups. Figure 5.8 shows the fraction of cells in each Leiden cluster from the different sequencing batches (Figure 5.8 B) and different treatment time points (Figure 5.8 A). As expected, cells from similar treatment time points cluster into the same clusters across sequencing batches and donors. This is also true for cells originating from the CAR T product (infusion product (IP)), which is expected to show similar gene expression across donors. Given that the clustering was better explained by experimental

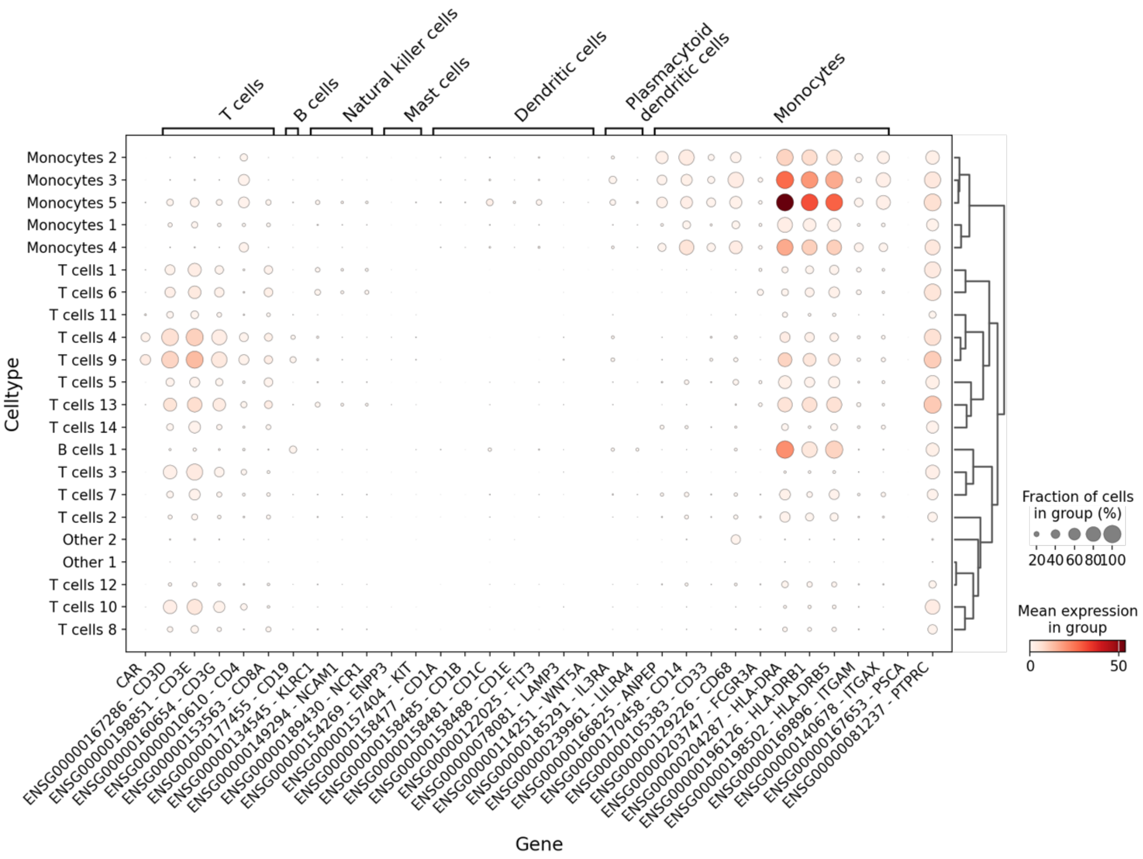


Figure 5.9 Expression of cell type marker genes in each Leiden cluster labeled with the corresponding cell type.

condition than sequencing batch, we decided to continue the analysis without batch correction to avoid the biases introduced by batch correction.

The Leiden clusters can subsequently be assigned cell types based on the expression of marker genes. Figure 5.9 shows the expression of cell type marker genes per Leiden cluster obtained without batch correction labeled with the corresponding cell type assignment. The clear separation of clusters by cell type marker genes and the similarity in gene expression space between clusters of the same cell type (see the dendrogram in Figure 5.9) further indicates that the clustering captured biological rather than batch effects. As expected, only T cells expressed the CAR transgene. Moreover, the presence of the CAR transgene correlated with the expression of CD19, which is not usually expressed in T cells. These CD19 counts likely originated from the CAR transgene, which also contains a copy of human CD19, rather than corresponding to endogenous expression of the CD19 gene as observed in B cells (Figure 5.9).

Comparison of gene expression results between donors

Using an approach similar to the assignment of cell types to Leiden clusters based on cell type marker genes, as discussed above, T cells can be further broken down into immune

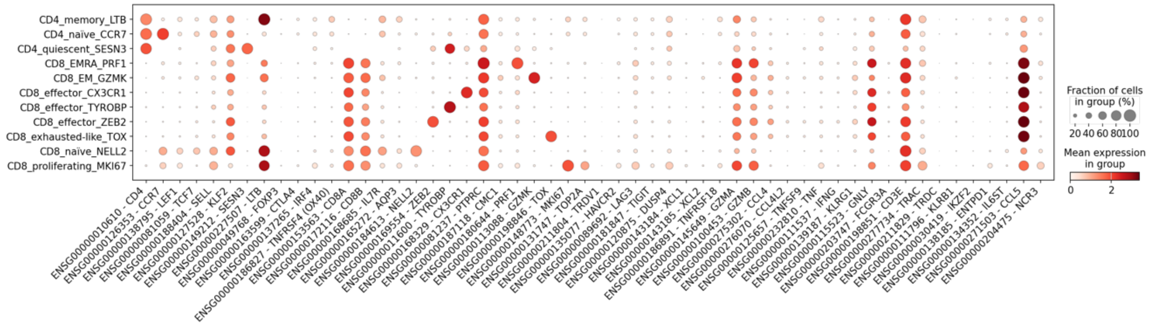


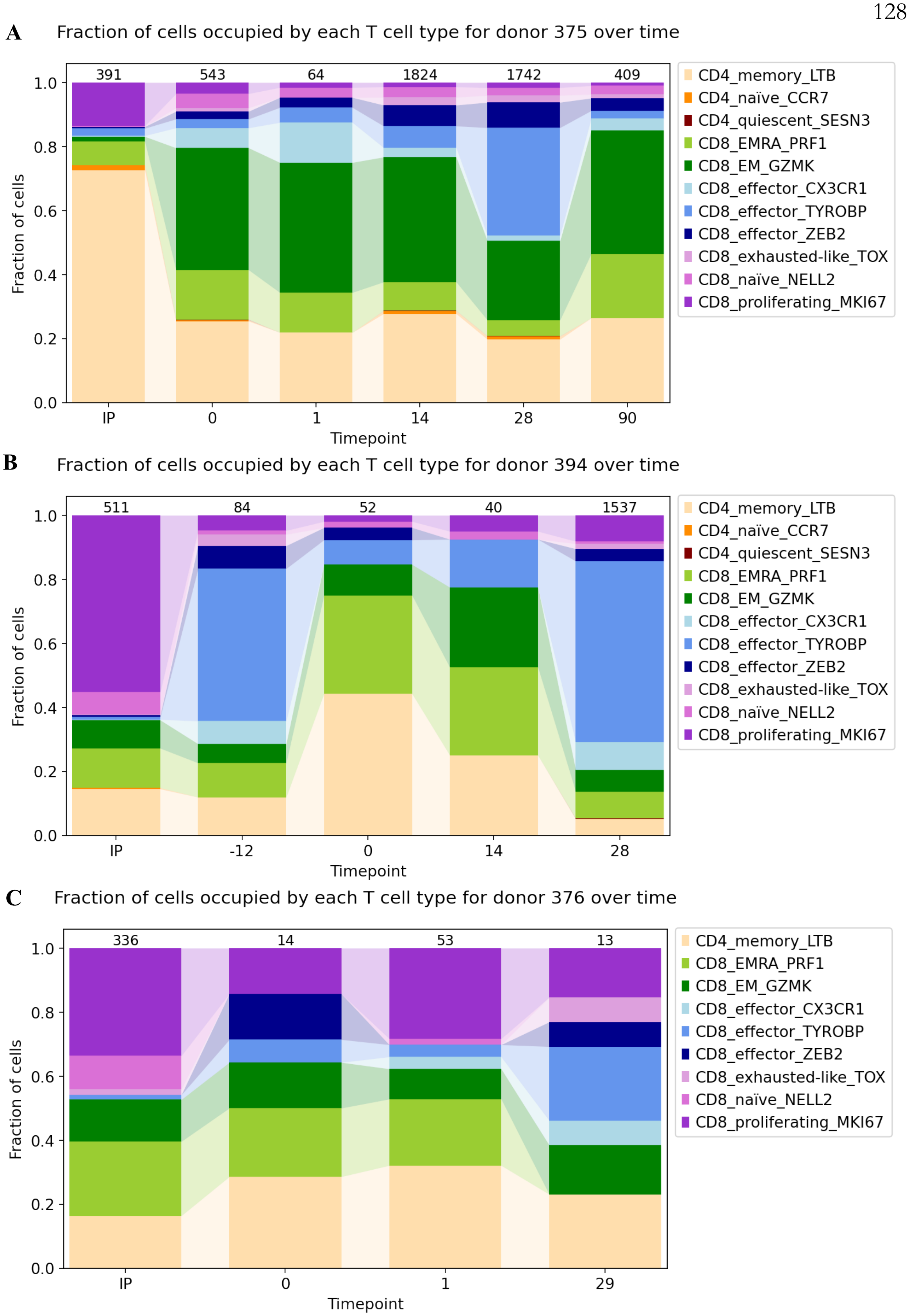
Figure 5.10 Expression of immune sub-cell type marker genes in groups of different T cell types.

sub-cell types based on marker genes for different types of T cells. Instead of re-clustering the T cells and assigning sub-cell types to smaller Leiden clusters, we assigned the sub-cell type labels to each individual T cell using a custom reiterative algorithm that increases marker gene expression thresholds with each round of sub-cell type assignment. This resulted in groups of immune sub-cell types with clearly defined expression of the corresponding marker genes (Figure 5.10).

This sub-cell type assignment allowed us to follow different populations of immune cells over time, as is shown for donor 388 in Figure 5.3f in the previous subchapter. The comparison of these results across several donors is hindered by the data heterogenicity between donors, partly caused by different medical histories, and the uneven distribution of time point data per donor (Figure 5.8 C). However, we hypothesized that global trends, such as changes in the fraction of cells occupied by each immune sub-cell type, may be reproducible across donors. Figure 5.11 shows the fraction of T cells occupied by each immune sub-cell type for donors 375, 394, and 376 over time. Although the number of available T cells per donor differs between time points, an expansion of effector CD8+ T cell subsets can be observed for all patients between 0- and 29-days post-infusion. This coincides with the results presented for donor 388 in Figure 5.3f in the previous subchapter.

Analysis of the V(D)J immune repertoire across donors

The expression of antigen binding domains from the V(D)J immune repertoire sequencing data was quantified using the V(D)J algorithm of 10x Genomics Cell Ranger v7.1.0. Interestingly, the expansion of clonotypes 28 days post-infusion, as observed for donor 388 (Figure 5.3g in the previous subchapter), could not be reproduced for donor 375 (Figure 7A). Figure 5.12 B shows the number of clonotypes contained in different fractions of cells at 28 days post-infusion, further visualizing that the clonotypes in donor 388 are split into dominating clonotypes present in large fractions of cells and fewer clonotypes occupying small fractions of cells. By contrast, the fractions of cells occupied by clonotypes in donor 375 are more evenly distributed (Figure 5.12 B).



A Fraction of cells occupied by each T cell type for donor 375 over time

B Fraction of cells occupied by each T cell type for donor 394 over time

C Fraction of cells occupied by each T cell type for donor 376 over time

Figure 5.11 Fraction of T cells occupied by each immune sub-cell type for donors 375 (**A**), 394 (**B**), and 376 (**C**) over time. The numbers above each bar delineate the total number of T cells at that time point.

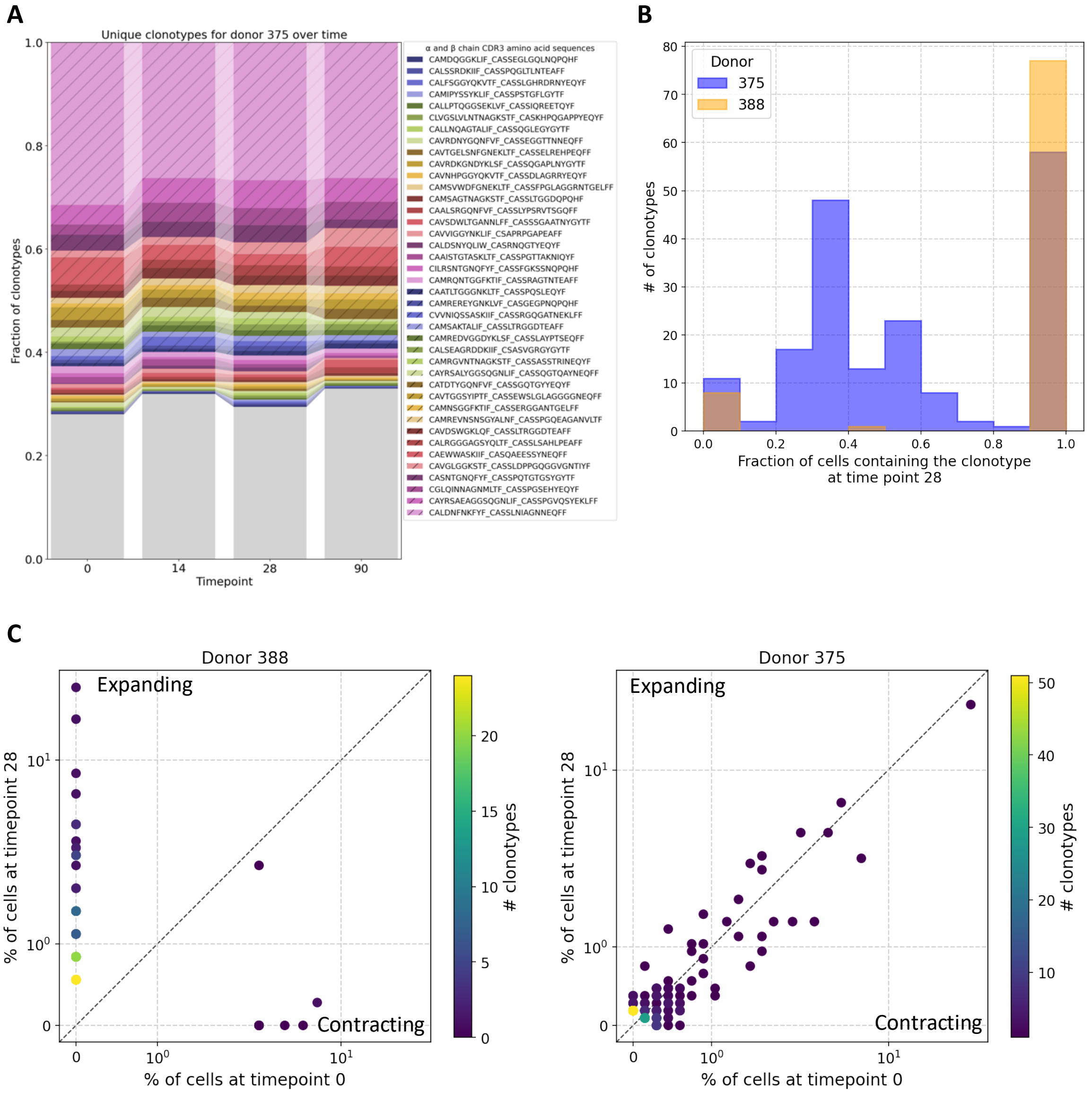


Figure 5.12 A The fraction of clonotypes occupied by individual clonotypes over time in the peripheral blood of donor 375. **B** The number of clonotypes occupying different fractions of cells for donors 375 and 388 at 28 days post-infusion. **C** The percentage of cells occupied by each individual clonotype at 28 days post-infusion over the percentage at 0 days post-infusion for donors 388 (left) and donor 375 (right).

In contrast to donor 388, donor 375 did not undergo lymphodepletion prior to infusion with CAR T product, which potentially prevented individual evolving clonotypes from inhabiting large fractions of the overall clonotype population as seen in donor 388. To investigate whether there were any expanding clonotypes over time in donor 375, we visualized the percentage of cells each clonotype was expressed at 0- and 28-days post-infusion (Figure 5.12 C). In this plot, expanding clonotypes are underrepresented at 0 days post-infusion, while contracting clonotypes are underrepresented at 28 days post-infusion.

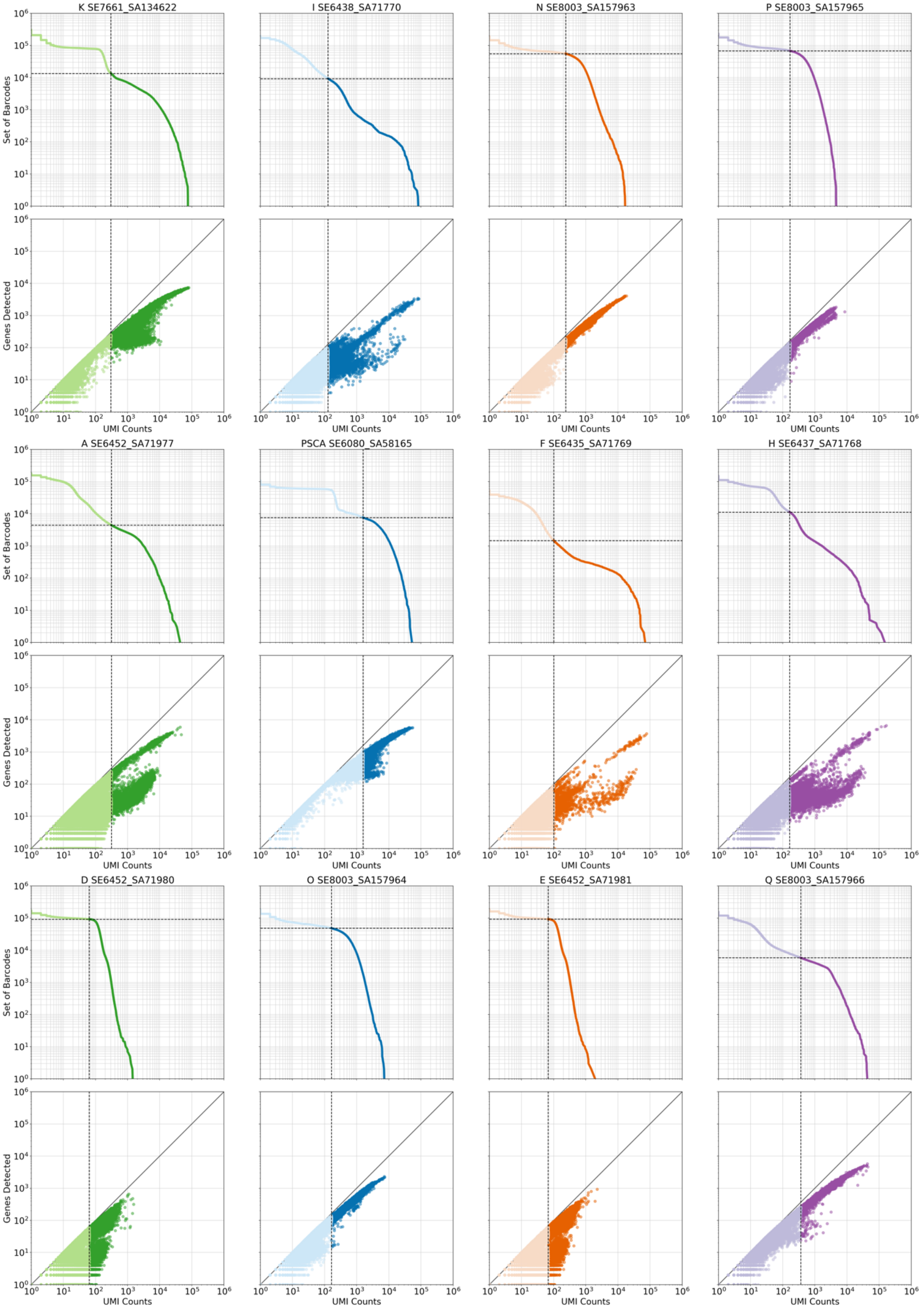
Donor 388 showed two separate populations of expanding and contracting clonotypes, with only a single clonotype remaining stable over time. In contrast, all clonotypes in donor 375 remained stable with only slight changes in the percentage of cells occupied by each clonotype between 0- and 28-days post-infusion, suggesting that donor 375 did not show any expansion of clonotypes.

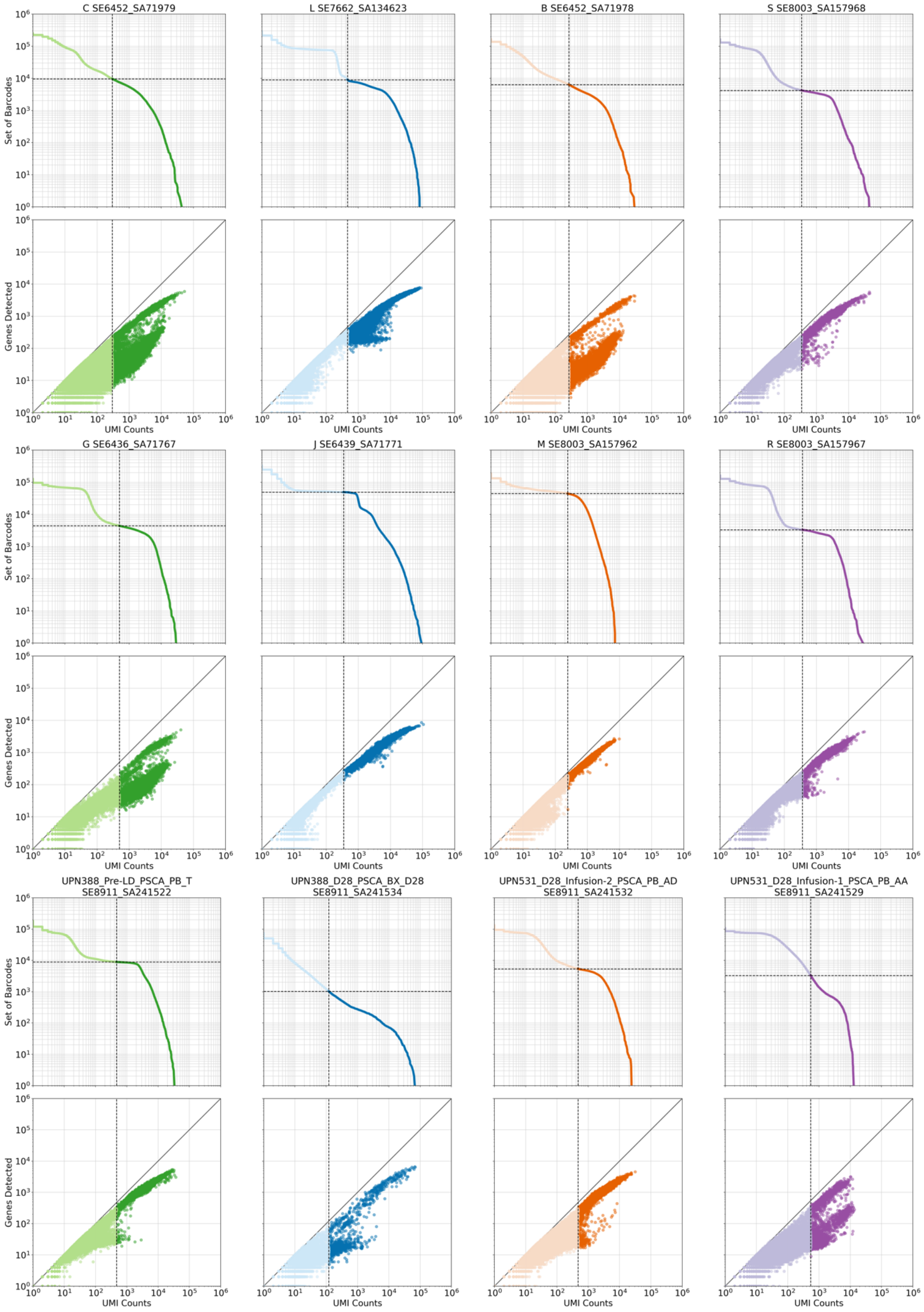
Code availability

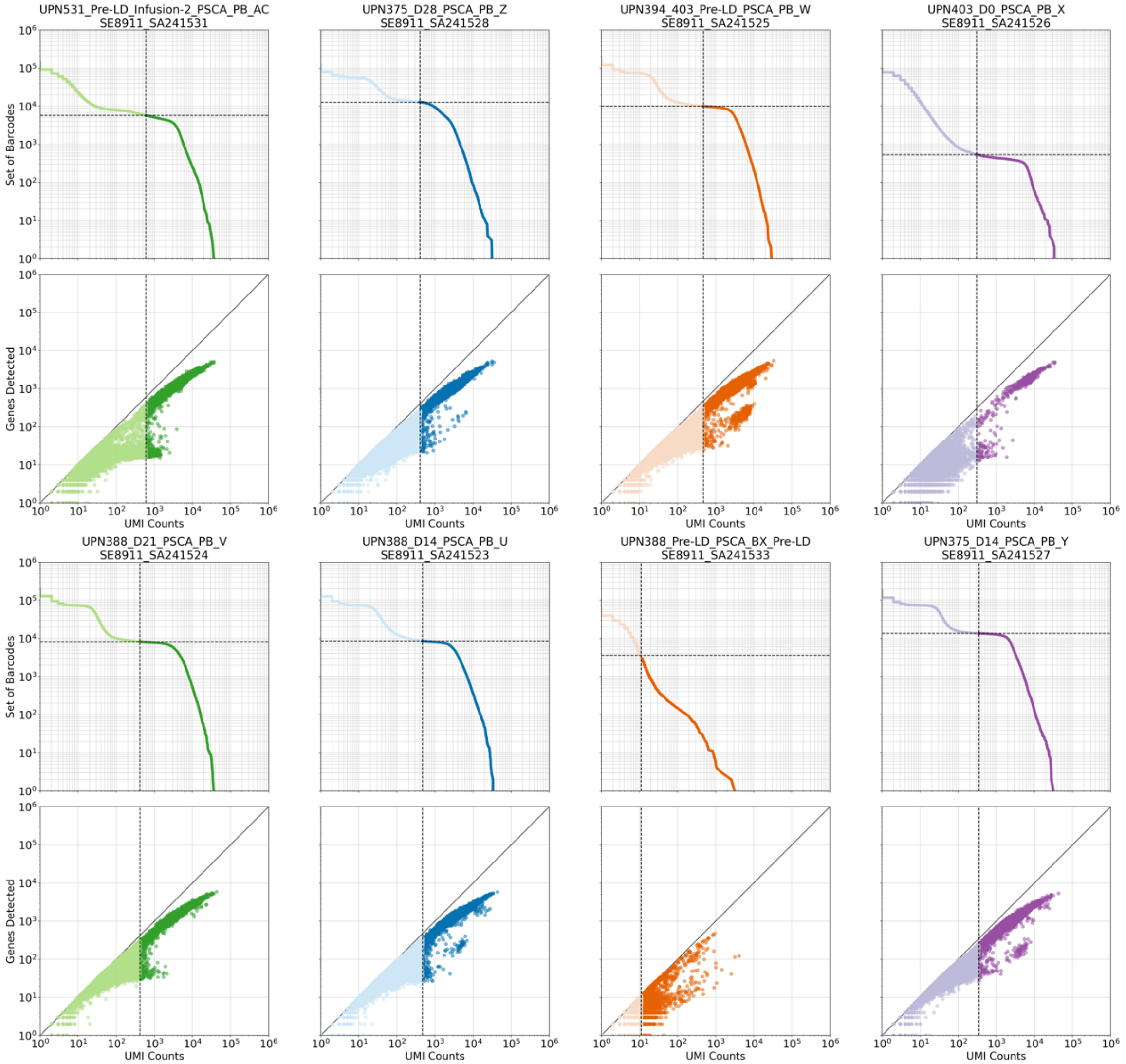
The code to reproduce the figures and analysis, as well as supplementary methods, can be found here: https://github.com/pachterlab/DBALLSMRDMCMGWSTPMBDKPFP_2023

References

1. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
2. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: Guaranteeing well-connected communities. *Sci. Rep.* **9**, 1–12 (2019).


K SE7661_SA134622
I SE6438_SA71770
N SE8003_SA157963
P SE8003_SA157965
A SE6452_SA71977
PSCA SE6080_SA58165
F SE6435_SA71769
H SE6437_SA71768
D SE6452_SA71980
O SE8003_SA157964
E SE6452_SA71981
Q SE8003_SA157966
Set of Barcodes
Genes Detected
UMI Counts
10^0
10^1
10^2
10^3
10^4
10^5
10^6


C SE6452_SA71979
L SE7662_SA134623
B SE6452_SA71978
S SE8003_SA157968
G SE6436_SA71767
J SE6439_SA71771
M SE8003_SA157962
R SE8003_SA157967
UPN388_Pre-LD_PSCA_PB_T
SE8911_SA241522
UPN388_D28_PSCA_BX_D28
SE8911_SA241534
UPN531_D28_Infusion-2_PSCA_PB_AD
SE8911_SA241532
UPN531_D28_Infusion-1_PSCA_PB_AA
SE8911_SA241529
Set of Barcodes
Genes Detected
UMI Counts
10^0
10^1
10^2
10^3
10^4
10^5
10^6



Extended Data Figure 5.1 Knee and library saturation plots for 32 gene expression datasets derived from prostate cancers patients at different time points of treatment with CAR T therapy. The dashed grey lines indicate the data quality threshold for barcode filtering.

Chapter 6

CONCLUSION AND OUTLOOK

While the development of hardware and wet-lab technologies for the high-throughput capture of transcriptomic and proteomic information surges ahead, computational methods for analyzing high-dimensional omics data are still catching up. We have only begun to understand the various sources of technological and biological noise in transcriptomics data and how to model them accurately[1-6]. Moreover, the refinement of algorithms for crucial analysis steps, such as data visualization, is ongoing–in some cases, this is true for already widely used algorithms[7].

The host sequence masking strategies described in Chapter 3 exemplify how seemingly minor decisions in the data analysis, in this case pertaining to the data alignment and the identification of host versus viral sequences, can significantly impact biological interpretations. These intricacies underscore the need to merge extensive knowledge from both biological research fields and computer science when analyzing transcriptomics data.

As the required fields of expertise continue to expand, rigorous analysis often necessitates collaborative efforts. Successful collaboration between biologists and computer scientists requires a mutual understanding of each other's challenges, fostering effective and respectful communication through a shared language. Moreover, software developers can enhance the usability of their software by following the guidelines discussed in Chapter 1, emphasizing accessibility, reproducibility, and relevance for users from diverse backgrounds.

Successful collaboration across disciplines unlocks new potential within existing technologies, as exemplified in Chapter 3 with the detection of previously unknown viral sequences from existing data. The findings described in this thesis were only possible through combining expertise from many fields of study, and the need for collaboration will only increase in the future.

Finally, computational biology, like all fields of science, needs more than just brilliant minds. A brilliant mind will fade in an unsupportive environment. As scientists, one of the most essential parts of our job is educating and mentoring the next generation of scientists. This is achieved through teaching, as well as actively creating a safe, stable, respectful, and comfortable institute-wide environment that empowers and encourages all aspiring scholars to try, experiment, not know (yet), learn, fail, and succeed. To quote Einstein: "I never teach my pupils. I only attempt to provide the conditions in which they can learn." If we let arrogance and ego get in the way of creating such conditions, we inevitably fail as scientists.

References

1. Gorin, G., Vastola, J. J., Fang, M. & Pachter, L. Interpretable and tractable models of transcriptional noise for the rational design of single-molecule quantification experiments. *Nat. Commun.* **13**, (2022).
2. Gorin, G. & Pachter, L. Length biases in single-cell RNA sequencing of pre-mRNA. *Biophys. Reports* **3**, 100097 (2023).
3. Gorin, G. & Pachter, L. Special function methods for bursty models of transcription. *Phys. Rev. E* **102**, 1–8 (2020).
4. Gorin, G. & Pachter, L. Intrinsic and extrinsic noise are distinguishable in a synthesis – export – degradation model of mRNA production. *bioRxiv* 2020.09.25.312868 (2020).
5. Carilli, M., Gorin, G., Choi, Y., Chari, T. & Pachter, L. Mechanistic modeling with a variational autoencoder for multimodal single-cell RNA sequencing data. *bioRxiv* 2023.01.13.523995 (2023).
6. Chari, T. *et al.* Whole animal multiplexed single-cell RNA-seq reveals plasticity of clytia medusa cell types. *bioRxiv* 2021.01.22.427844 (2021).
7. Chari, T. & Pachter, L. The specious art of single-cell genomics. *PLoS Comput. Biol.* **19**, 1–20 (2023).