*Bioinformatics Advances*, 2023, vbad071
https://doi.org/10.1093/bioadv/vbad071
Advance access publication 9 June 2023
**Application Note**

OXFORD

Software

# tidytof: a user-friendly framework for scalable and reproducible high-dimensional cytometry data analysis

**Timothy J. Keyes [1,2,3], Abhishek Koladiya[3], Yu-Chen Lo[3], Garry P. Nolan[4] and Kara L. Davis [3,5,*]**

[1]Medical Scientist Training Program, Stanford University School of Medicine, Stanford, CA 94305, USA
[2]Department of Biomedical Data Science, Stanford University School of Medicine, Stanford, CA 94305, USA
[3]Department of Pediatrics, Stanford University School of Medicine, Stanford, CA 94305, USA
[4]Department of Pathology, Stanford University School of Medicine, Stanford, CA 94305, USA
[5]Center for Cancer Cell Therapy, Stanford University School of Medicine, Stanford, CA 94305, USA

*To whom correspondence should be addressed.

Associate Editor: Guoqiang Yu

## Abstract

**Summary:** While many algorithms for analyzing high-dimensional cytometry data have now been developed, the software implementations of these algorithms remain highly customized—this means that exploring a dataset requires users to learn unique, often poorly interoperable package syntaxes for each step of data processing. To solve this problem, we developed {tidytof}, an open-source R package for analyzing high-dimensional cytometry data using the increasingly popular 'tidy data' interface.

**Availability and implementation:** {tidytof} is available at https://github.com/keyes-timothy/tidytof and is released under the MIT license. It is supported on Linux, MS Windows and MacOS. Additional documentation is available at the package website (https://keyes-timothy.github.io/tidytof/).

**Contact:** kardavis@stanford.edu

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

## 1 Introduction

Over the past decade, high-dimensional cytometry has become a prominent technology for high-throughput single-cell analysis of both human and animal tissues (Spitzer and Nolan, 2016). Platforms including mass cytometry (or Cytometry by Time-Of-Flight), full-spectrum flow cytometry and sequence-based cytometry have now enabled the collection of large datasets of multiplexed proteomic measurements from millions of cells per experiment (Bendall *et al.*, 2011; Jaimes *et al.*, 2022; Keyes *et al.*, 2020; Salomon *et al.*, 2020). To derive insights from these complex datasets, recent years have also seen the development of dozens of algorithms for analyzing high-dimensional cytometry data at the single-cell, cell subpopulation and whole-sample levels (Fig. 1A) [for comprehensive reviews, see Keyes *et al.* (2020), Olsen *et al.* (2019) and Hu *et al.* (2022)]. However, navigating the selection, use and interoperability requirements for each of these methods remains a significant challenge for many biologists.

Within the same time-period, the concept of 'tidy data'—i.e. data represented in flexible, two-dimensional tables (called *data frames*) in which each row represents an observation and each column represents an experimental variable—has constituted a paradigm shift within the field of data science (Fig. 1 and Supplementary Table S1) (Wickham, 2014). The central concept of data 'tidiness' is that representing data in a consistent, tidy format makes data processing simpler (and often faster) by standardizing the tools needed to build an analytical pipeline. As such, adopting tidy data practices generally encourages the use of intuitive, human-centered design principles in statistical software engineering, allowing researchers to apply similar analytical frameworks across tools and research domains by expressing common data processing operations using a self-consistent vocabulary. In the context of scientific computing, using tidy data is particularly powerful because it results in code that is easy-to-read (and thereby easy-to-maintain), making it error-resistant, reproducible and relatively easy to write and debug. Because of these useful qualities, tidy workflows are emerging for bioinformatics applications in genomic (Lee *et al.*, 2019) and transcriptomic analysis (Mangiola *et al.*, 2021a, b). Here, we build on this previous work by presenting {tidytof}, an R package implementing a comprehensive, performant and extensible framework for analyzing high-dimensional cytometry data with a tidy interface. Comprehensive documentation and tutorials for how to use {tidytof} are available in Supplementary Notes S1 and S2 and in the {tidytof} package vignettes.

**Received:** January 20, 2023. **Revised:** May 3, 2023. **Editorial Decision:** May 27, 2023. **Accepted:** June 7, 2023

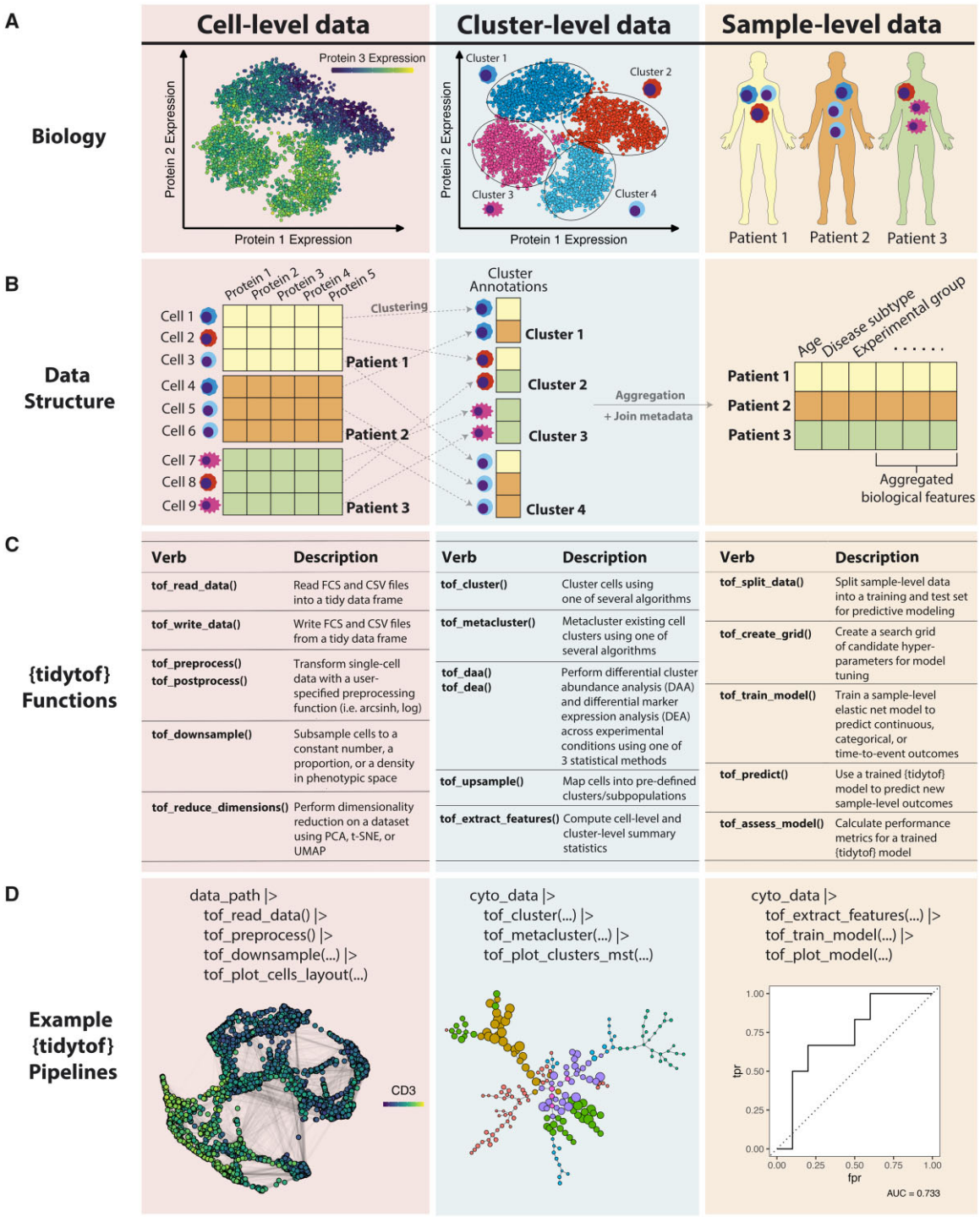


**Figure 1.** {tidytof} implements a 'grammar' of single-cell data analysis at the single-cell, cluster and whole-sample levels. (**A**) Mass cytometry data contains biological insights at multiple levels of observation. The cell level (left) contains information about individual cells' marker expression profiles. The cluster level (middle) contains information about the cell subpopulations that predominate among single-cell phenotypes. The whole-sample level (right) contains information about patient outcomes or experimental conditions that can be associated with single-cell and cluster-level information. (**B**) 'Tidy' data structures can be used to represent high-dimensional cytometry data at the single-cell, cluster and sample levels. At the cell level (left), a tidy data frame represents each cell as a row and each protein measurement as a column (each square in the image represents a single numeric measurement). Cells from multiple samples can be grouped by their sample-of-origin (represented using the color of each row) and bound row-wise into a single tidy data frame (see Supplementary Table S1 for an expanded representation). At the cluster level (middle), cluster membership can be represented by assigning a cluster identifier to each cell, allowing cells to be regrouped according to cluster membership. Finally, information from the single-cell and cluster levels can then be aggregated by computing summary statistics (such as each cluster's abundance and mean marker expression) and joined with sample-level metadata to form a tidy data frame in which each row represents a sample and each column represents an aggregated biological feature or piece of metadata (right). (**C**) {tidytof} functions associated with the cell, cluster and whole-sample levels of data analysis. For expanded versions of these tables, see Supplementary Tables S2–S4. (**D**) Example code snippets that use {tidytof} functions to process and visualize high-dimensional cytometry data at the cell (left), cluster (middle) and whole-sample (right) levels. {tidytof} functions are chained together into mix-and-match data processing pipelines using R's native pipe operator ('|>'). In the snippets, 'data_path' represents a file path to input .fcs or .csv files, 'cyto_data' represents a high-dimensional cytometry dataset in tidy format, and . . . represents function-specific arguments abbreviated for clarity. See Supplementary vignettes for additional details

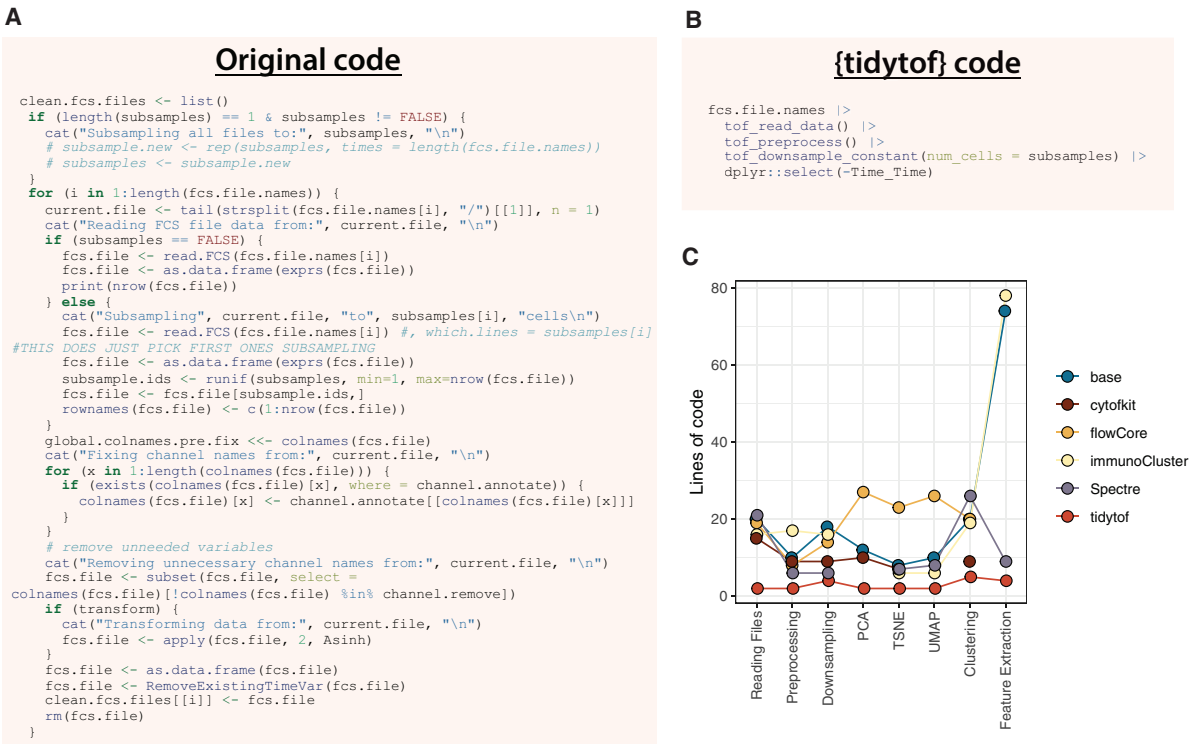


**Figure 2.** {tidytof} simplifies the code needed to perform high-dimensional cytometry data analysis. (**A**) Raw code from the body of LoadCleanFCS(), a function used for reading, preprocessing and downsampling FCS files in Ko *et al.* (2020). This code is replicated without editing, and the original source is available here: https://github.com/zunderlab/FLOWMAP/blob/master/R/load-from-FCS.R. Note that in the original function, the variable *fcs.file.names* is a character vector containing the file path to each FCS file to be read into memory, and the variable *subsamples* is an integer indicating how many cells to samples from each FCS file. (**B**) {tidytof} code that provides the same functionality as that provided in (**A**). *fcs.file.names* and *subsamples* are variables defined as in (A). (**C**) Raw lines of code needed to perform equivalent analyses using {tidytof} and other open-source single-cell data analysis packages, including base R, {cytofkit}, {flowCore}, {immunoCluster} and {Spectre}. Functions used to perform the code comparisons are provided at the following GitHub link: https://github.com/keyes-timothy/tidytof-manuscript/blob/72c2d347241660156b1751cfdad1f0849df3dccb/benchmarking/benchmarking_functions.R. For comprehensive coding burden comparisons with other packages, see Supplementary Figures S3 and S4

## 2 Software design

{tidytof} consolidates popular methods of high-dimensional cytometry data analysis, such as file reading, preprocessing, batch correction, quality control, clustering, dimensionality reduction, differential expression analysis, feature extraction and visualization into a composable, easy-to-use application programming interface (API) suitable for both experienced and beginner programmers (see Supplementary Tables S2–S4). It does so by organizing common data analysis tasks—including state-of-the-art statistical analyses and machine-learning algorithms—into verbs, or modular function families that represent specific mathematical operations like dimensionality reduction and clustering (Fig. 1C). Each {tidytof} verb accepts user commands with the same structure: a tidy data frame is accepted as input, and a tidy data frame is returned as output. Specifically, {tidytof} stores high-dimensional cytometry data in a specialized data structure called a *tof_tbl* (Supplementary Table S1). *tof_tbl*'s are an extension of R's *data.frame* class that are customized to store cytometry data in a tidy format: cells are represented as rows, protein (or transcript) measurements are represented as columns and metadata about the cytometry panel used during data acquisition is stored as a class attribute. In addition, *tof_tbl*'s can also incorporate columns (using tidytof verbs' *augment* flag) that store additional information about each cell—e.g. a cluster assignment, a low-dimensional embedding, or metadata information about the sample from which the cell was collected. Unlike other data structures commonly used to store single-cell data (such as *SingleCellExperiment* or *flowSet* objects), *tof_tbl*'s directly inherit behavior from *data.frame*'s, enabling them to dispatch any of the many existing methods for the *data.frame* class (Amezquita *et al.*, 2020; Ellis *et al.*, 2021). Furthermore, a key feature of {tidytof}'s API is its ability to assemble multiple functions into modular, multistep pipelines that combine several analytical steps but remain easily human-readable by iteratively transforming a *tof_tbl* one step at a time (Fig. 1D and Supplementary Notes S1 and S2). Thus, tidytof stores cell-level data in the rows of a *tof_tbl*, allowing cluster- and sample-level data to be quickly aggregated using {tidytof} verbs' grouped operations.

This design strategy has three major benefits. First, it allows {tidytof} to simplify how users access algorithms that perform the same fundamental task in different ways. For example, the *tof_cluster* verb provides a single interface to multiple clustering algorithms (see Supplementary Table S3) that, despite their distinct underlying implementations, can all be applied easily using {tidytof}'s consistently tidy input and output data formats. This means that users can focus their efforts on interpreting and refining the results of a data analysis pipeline rather than on maintaining large code bases, which {tidytof} can simplify into relatively few lines of code (Fig. 2). Second, {tidytof} verbs' modular design means that they can be combined to perform flexible analyses at each level of biological scope inherent to single-cell data: the single-cell level, the cell subpopulation level and the whole-sample level. {tidytof} provides verbs for computation and

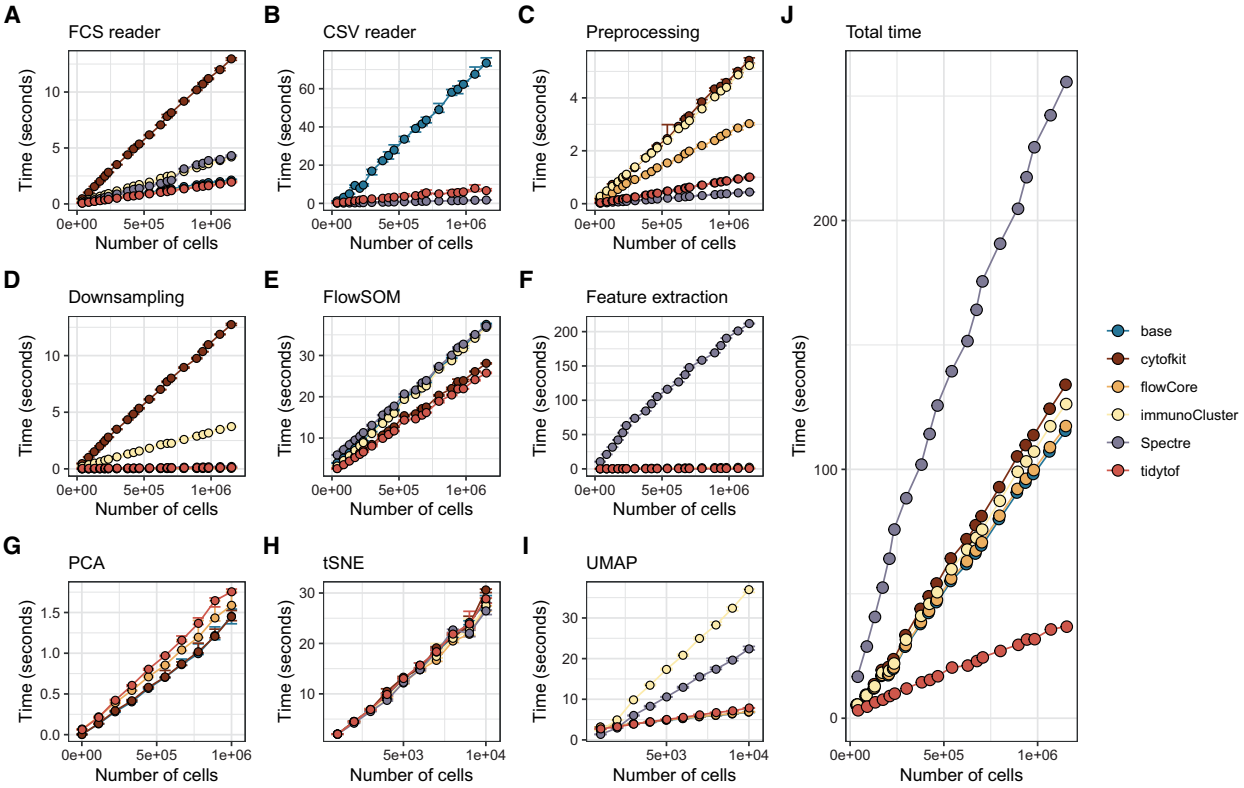


**Figure 3.** {tidytof}'s computational speed rivals or improves upon equivalent approaches using existing tools. Benchmark plots indicating the elapsed time for (**A**) reading FCS files, (**B**) reading CSV files, (**C**) preprocessing single-cell data, (**D**) downsampling single-cell data, (**E**) performing FlowSOM clustering, (**F**) sample-level feature extraction, (**G**) performing principal component analysis (PCA), (**H**) performing *t*-stochastic neighborhood embedding (tSNE), or (**I**) performing UMAP embedding using {tidytof}, base R, {flowCore}, {cytofkit}, {immunoCluster} and {Spectre}. Panel (**J**) shows the sum of runtimes from panels (A–F) for each tool [because panels (G–I) were run on different datasets, they could not be combined with the other panels] In all panels, points represent the median runtime for each workflow from 10 independent repetitions and error bars represent the interquartile range of runtimes

visualization within each of these levels as well as for transforming data between them. This results in a concise and intuitive domain-specific language for single-cell data analysis designed to answer questions at multiple levels of biological interest (Fowler, 2010). Third, as an extension of the high-performing tidyverse ecosystem of data analysis tools in R, {tidytof} provides superior computational performance relative to existing packages for single-cell data analysis.

## 3 Performance benchmarking

We benchmarked {tidytof}'s speed and memory performance against equivalent workflows in two low-level APIs (base R and {flowCore}) as well as three high-level APIs ({cytofkit}, {immunoCluster} and {Spectre}) with similar capabilities (Ashhurst *et al.*, 2021; Chen *et al.*, 2016; Ellis *et al.*, 2021; Opzoomer *et al.*, 2021; R Core Team, 2021). Workflows—which included reading input files, preprocessing, downsampling, clustering, dimensionality reduction and feature extraction—were evaluated on the basis of their computational speed and memory performance on datasets sampled from a publicly available cohort of mass cytometry samples (Good *et al.*, 2018). Across all workflows, {tidytof}'s computational performance rivaled or surpassed existing tools (Fig. 3A–I), and {tidytof}'s total time across all workflows was the smallest for all datasets used for benchmarking (Fig. 3J).

Notably, {tidytof}'s competitive performance is primarily attributable to efficient usage of the tidyverse package ecosystem, which interfaces easily with many other packages as an extension of the base R environment. This is exemplified by {tidytof}'s superior performance at file reading and downsampling compared to {cytofkit}, whose non-modular design results in redundant computations; by its superior performance at feature extraction compared to {Spectre}, which suffers a substantial performance penalty due to its use of for-loops; and its superior wrapping of uniform manifold approximation and projection (UMAP) compared to {immunoCluster} and {Spectre}, both of which are substantially slowed by their need to convert to and from *data.table* and *SingleCellExperiment* data objects, respectively. Despite this, {tidytof} is also as memory efficient as other tools for high-dimensional cytometry data analysis, suggesting that its speed does not derive from a tradeoff between data storage and computational performance (Supplementary Fig. S2). For comprehensive details and additional benchmarking, see the '{tidytof} performance benchmarking' section of the Supplementary Material.

## 4 Conclusion

In summary, {tidytof} provides a tidy interface for analyzing high-dimensional cytometry data with an easy-to-use API and seamless integration with many existing tools created by the data science and bioinformatics communities (Wickham *et al.*, 2019). In doing so, {tidytof} decreases the coding burden of applying standard tools to high-dimensional cytometry

datasets, thereby increasing the accessibility of advanced analytical methods to researchers with limited programming experience. While explicitly focused on cytometry data, {tidytof} also proposes a more general framework for creating a unified 'grammar' of single-cell data analysis—one that involves standard strategies for reading/writing data, preprocessing, clustering, feature engineering/aggregation and statistical modeling. This framework is widely extensible beyond cytometric technologies themselves. Our future engineering efforts will include both implementing new data processing features as additional verbs as well as extending existing verbs' compatibility with other emerging single-cell technologies, such as high-dimensional imaging (Black *et al.*, 2021; Levenson *et al.*, 2015; Ståhl *et al.*, 2016).

## Acknowledgements

We thank Astraea Jager, Brooks Benard, Caterina Colón, Jeremy D'Silva, Jolanda Sarno, Pablo Domizi and the R/Medicine community for helpful discussions.

## Author contributions

Timothy Keyes (Conceptualization [lead], Data curation [lead], Formal analysis [lead], Funding acquisition [lead], Investigation [lead], Methodology [lead], Project administration [lead], Resources [lead], Software [lead], Validation [lead], Visualization [lead], Writing—original draft [lead], Writing—review & editing [lead]), Abhishek Koladiya (Data curation [equal], Formal analysis [equal], Investigation [equal], Methodology [equal], Validation [equal], Visualization [equal], Writing—review & editing [equal]), Yu-Chen Lo (Formal analysis [equal], Investigation [equal], Methodology [equal], Validation [equal], Visualization [equal], Writing—review & editing [equal]), Garry Nolan (Conceptualization [equal], Funding acquisition [equal], Investigation [equal], Project administration [equal], Resources [equal], Supervision [equal], Writing—original draft [equal], Writing—review & editing [equal]), and Kara L Davis (Conceptualization [equal], Data curation [equal], Formal analysis [equal], Funding acquisition [lead], Investigation [equal], Methodology [equal], Project administration [equal], Resources [equal], Supervision [equal], Validation [equal], Writing—original draft [equal], Writing—review & editing [equal])

## Funding

This work was supported by the National Institutes of Health [grant numbers 1F31CA239365-01 to T.J.K., U54CA 209971, U54HG010426, U19AI100627 to G.P.N. and R01 CA251858-01A1 to K.L.D.]; the Stanford Maternal and Child Health Research Institute; the Mark Foundation for Cancer Research; the Andrew McDonough B+ (Be Positive) Foundation; and a Point Foundation Graduate Student Scholarship (to T.J.K.).

## Conflict of Interest

None declared.

## Data availability

Mass cytometry data used for performance benchmarking are available at https://github.com/kara-davis-lab/DDPR/releases.

## References

Amezquita,R. *et al.* (2020) Orchestrating single-cell analysis with Bioconductor. *Nat. Methods*, **17**, 137–145.

Ashhurst,T.M. *et al.* (2021) Integration, exploration, and analysis of high-dimensional single-cell cytometry data using Spectre. *Cytometry A*, **101**, 237–253.

Bendall,S.C. *et al.* (2011) Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, **332**, 687–696.

Black,S. *et al.* (2021) CODEX multiplexed tissue imaging with DNA-conjugated antibodies. *Nat. Protoc.*, **16**, 3802–3835.

Chen,H. *et al.* (2016) Cytofkit: a bioconductor package for an integrated mass cytometry data analysis pipeline. *PLoS Comput. Biol.*, **12**, e1005112.

Ellis,B. *et al.* (2021) flowCore: basic structures for flow cytometry data.

Fowler,M. (2010) *Domain-Specific Languages*. Addison-Wesley, Upper Saddle River, NJ.

Good,Z. *et al.* (2018) Single-cell developmental classification of B cell precursor acute lymphoblastic leukemia at diagnosis reveals predictors of relapse. *Nat. Med.*, **24**, 474–483.

Hu,Z. *et al.* (2022) Application of machine learning for cytometry data. *Front. Immunol.*, **12**, 787574.

Jaimes,M.C. *et al.* (2022) Full spectrum flow cytometry and mass cytometry: a 32-marker panel comparison. *Cytometry A*, **101**, 942–959.

Keyes,T.J. *et al.* (2020) A cancer biologist's primer on machine learning applications in high-dimensional cytometry. *Cytometry A*, **97**, 782–799.

Ko,M.E. *et al.* (2020) FLOW-MAP: a graph-based, force-directed layout algorithm for trajectory mapping in single-cell time course datasets. *Nat. Protoc.*, **15**, 398–420.

Lee,S. *et al.* (2019) plyranges: a grammar of genomic data transformation. *Genome Biol.*, **20**, 4. https://doi.org/10.1186/s13059-018-1597-8.

Levenson,R.M. *et al.* (2015) Immunohistochemistry and mass spectrometry for highly multiplexed cellular molecular imaging. *Lab. Invest.*, **95**, 397–405.

Mangiola,S. *et al.* (2021a) Interfacing Seurat with the R tidy universe. *Bioinformatics*, **37**, 4100–4107.

Mangiola,S. *et al.* (2021b) tidybulk: an R tidy framework for modular transcriptomic data analysis. *Genome Biol.*, **22**, 42. https://doi.org/10.1186/s13059-020-02233-7.

Olsen,L.R. *et al.* (2019) The anatomy of single cell mass cytometry data. *Cytometry A*, **95**, 156–172.

Opzoomer,J.W. *et al.* (2021) ImmunoCluster provides a computational framework for the nonspecialist to profile high-dimensional cytometry data. *Elife*, **10**, e62915.

R Core Team (2021) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Salomon,R. *et al.* (2020) Genomic cytometry and new modalities for deep single-cell interrogation. *Cytometry A*, **97**, 1007–1016.

Spitzer,M.H. and Nolan,G.P. (2016) Mass cytometry: single cells, many features. *Cell*, **165**, 780–791.

Ståhl,P.L. *et al.* (2016) Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, **353**, 78–82.

Wickham,H. (2014) Tidy data. *J. Stat. Softw.*, **59**, 1–23.

Wickham,H. *et al.* (2019) Welcome to the tidyverse. *JOSS*, **4**, 1686.