barbieQ: An R software package for analysing barcode count data from clonal tracking experiments

Liyang Fei [1,2], Jovana Maksimovic [1,2], Alicia Oshlack [1,2,3]
1 Peter MacCallum Cancer Centre, Melbourne, Victoria 3000, Australia
2 Sir Peter MacCallum Department of Oncology, The University of Melbourne, Victoria 3010, Australia
3 School of Mathematics and Statistics, The University of Melbourne, Victoria 3010, Australia

Abstract

Tracking the clonal composition of pools of cells as they differentiate or evolve can reveal important features of development and clonal evolution in fields such as stem cell research and cancer. A variety of new single-cell lineage tracing technologies now allow us to track progeny cells back to their original progenitor cell using unique DNA barcodes that are passed from each original cell to its offspring. The barcode count for each sample indicates the number of progeny cells derived from the original cell. However, as the different clonal tracking technologies often require tailored pipelines for barcode count calling, subsequent interpretation of barcode count data is often conducted through bespoke analyses, typically relying on visualisations and heuristics. A generalized workflow for preprocessing and robust statistical analysis of barcode count data across protocols enables studies to achieve the most biologically meaningful and reproducible results by rigorously accounting for sample-to-sample variability.

To address this, we introduce *barbieQ*, a Bioconductor R package designed for analysing barcode count data across groups of samples. *barbieQ* is built upon the widely used *SummarizedExperiment* data structure, making it interoperable with low-level sequence processing tools. It provides data-driven quality control and filtering, implements statistical analysis for testing between samples, and supports visualisations throughout the analysis. Two statistical tests included in *barbieQ* are 1) Differential barcode proportion that identifies barcodes with significantly changing proportions between groups of samples, and 2) Differential barcode occurrence that identifies barcodes more frequently seen in the samples of one condition compared to another.

barbieQ can handle complex experimental designs using regression models to test a factor of interest while accounting for other variables. Both tests maintain a type 1 error less than 5% under the null hypothesis using real-world and simulated data. Notably, in analysis of a real-world dataset of monkey blood stem cell differentiation, *barbieQ* effectively selected a larger set of informative barcodes and accurately identified novel $NK^{CD56-CD16+}$ cell-specific clones whilst avoiding less-specific barcodes, in contrast to the original analysis based on heuristic cutoffs and visualisations commonly used in the field. Overall, *barbieQ* is a powerful Bioconductor package with greater power to detect biologically meaningful changes leading to robust and reproducible analysis.

Keywords

lineage tracing, clonal tracking, bioinformatics, data analysis, single-cell, lentiviral barcoding, DE analysis

Introduction

Cellular DNA barcoding is a powerful technique for cell clonal tracking, also known as lineage tracing. This method uses unique synthetic DNA barcodes to genetically label each initial cell in a pool.[1] These heritable barcodes are then passed on to daughter cells, allowing for the identification of the origin of each progeny cell. Collectively, an initial cell and all its progeny are referred to as a “clone” or “lineage”. By tracking the behaviour of these clones, researchers can uncover clonal heterogeneity resulting from intrinsic differences among initial cells, such as variations in stem cell potency, and tumour cell heterogeneity.[2–5] This research method is often termed as “lineage tracing” in haematopoietic stem cell differentiation or “clonal tracking” in tumour cell evolution studies.[4–7]

While barcode incorporated in the genome serves as a unique identifier to distinguish between clones, researchers can track clones within a mixed population. Clonal size is the number of cells derived from each initial cell in a sample and can be determined by sequencing and counting the number of corresponding DNA barcodes, thereby also allowing relative clone abundance to be quantified. Based on the findings relating to clonal size and clonal abundance patterns, many studies subsequently look at the driving features of the initial cells, such as gene expression and gene activity patterns that relate to clonal outcome patterns.[2–5,8–10]

There are a variety of published protocols for cellular genetic barcoding, covering steps such as barcode sequence design, barcode-to-cell incorporation, and barcode sequencing.[4,11–15] These protocols have given rise to a series of in-house workflows, protocol-specific bioinformatics tools, and generalised software, for barcode sequence recovery, including extracting barcode sequences and correcting mismatched base pairs by aligning to the known barcode library or, alternatively, using library-free methods.[5,8,13,14,16–22] Furthermore, technologies used to capture barcode sequences also vary. While next-generation-sequencing (NGS) combined with barcode-specific PCR amplification is commonly used to efficiently capture barcode-specific DNA from bulk cell samples, studies aiming to capture both synthetic barcodes and cell mRNA, either in bulk or at the single-cell level, require different sequencing approaches. This adds complexity to barcode extraction and further necessitates tailored barcode preprocessing pipelines. In addition, *CellBarcodeSim* software can simulate barcode data with respect to the technical effects introduced throughout the barcoding processes.[22,23]

While there are various barcode extraction and alignment approaches, these pipelines eventually output a matrix of barcode read counts across samples. A barcode matrix typically contains thousands of barcodes across tens to hundreds of samples. Each barcode read represents the detection of a cell carrying that barcode in each sample, and the read count represents the number of cells derived from the originally barcoded cell, i.e. clone size. However, in studies such as *in vivo* haematopoietic stem cell tracking, not all progeny cells can be fully collected, so barcode counts reflect the relative abundance of each lineage in a sample. This is in contrast to studies such as *in vitro* tumour heterogeneity tracking, where progeny cells can be fully collected and sequenced, thereby allowing for barcode counts to be interpreted as absolute abundance. Although, in the second scenario, researchers may not sequence the entire repertoire of collected progeny cells or know the exact total cell count. Consequently, it's more common and practical to look at barcode count as relative or proportional abundance.

Several bioinformatics tools have been developed to analyse barcode count matrixes offering a range of visualization approaches, combined with heuristic interpretation strategies, specifically, the Bioconductor tool *barcodetrackR*, and GitHub hosted tools *CellDestiny* and *bartools*.[21,24–28] These tools share some common features:

All three tools visualise the detailed abundance of each barcode per sample, typically using heatmaps. *BarcodetrackR* and *bartools* display the log normalised barcode proportion or counts per million (CPM) for selected barcodes (top 10 abundant per sample), whilst *CellDestiny* displays the arcsine transformed proportion across all barcodes. Additionally, *CellDestiny* provides line plots and bubble plots, and *bartools* provides stacked bar plots and bubble plots for displaying barcode abundance across multiple samples.

All three tools summarise sample characteristics using the per-sample Shannon diversity index and number of features (detected unique barcodes per sample). To assess sample dissimilarities, *barcodetrackR* uses MDS plot and *bartools* uses PCA plots. Both assess sample similarities by displaying sample pair-wise correlations using the Pearson correlation coefficient. Notably, *CellDestiny* characterises the distribution of barcode abundance within each sample by visualising the cumulative barcode proportion per sample.

The three tools differ in how they characterize barcode change between samples. *barcodetrackR* calculates the log fold change (LFC) of barcode proportions between two groups of samples and categorizes barcodes according to a heuristic threshold. *CellDestiny* compares barcode relative abundance across groups by first applying column-wise and then row-wise normalisation to the barcode count matrix, then classifying the barcodes as either group-specific or universal based on whether each group passes a heuristic threshold. Neither of these tools provides a statistical test. *bartools* tests the odds of barcode presence/absence between two groups of progeny cells using a hypergeometric test, typically applied in the context of two clusters in single cell barcoding data.

Table 1. Feature comparison of existing tools for barcode count analysis

		barcodetrackR	CellDestiny	bartools
barcode abundance per sample	Visualisation	Heat map	Heat map Bubble plot Line plot	Heat map Bubble plot Stacked bar plot
	Normalization	Log proportion	Arcsine proportion	Log CPM
	Filtering	Top 10 most abundant	Rely on QC	Top 10 most abundant
Sample characteristics	Characteristics	Shannon Index Num. of features	Shannon Index Num. of features Distribution of barcode abundance per sample	Shannon Index Num. of features
	Dissimilarity	MDS plot		PCA plot
	Similarity	Sample pair Pearson correlation		Sample pair Pearson correlation
Barcode characteristics	Abundance	Compare LFC on proportion to a threshold	Compare row-normalised proportion to a threshold	
	Presence/absence			Hypergeometric test on odds of presence/absence

Lineage tracing experiments often involve complex experimental designs, including multiple time points, genotypes, treatments, and both biological and technical replicates. For example, in a mouse HSPC study, the clonal output was determined by a combination of the cell type cohorts, different tissues and various time points where cells were collected.[25] Despite this complexity, existing tools lack statistical frameworks capable of appropriately testing for differences between groups of samples.

1) The LFC approach, the relative proportion approach or the hypergeometric test cannot directly handle multiple conditions within a single factor.
2) None of the three approaches can robustly address within-group variability. Technical replicates can be aggregated in advance by *CellDestiny* and *bartools*, however, these approaches cannot rigorously address the difference within replicates and between experimental groups.
3) Both LFC and relative proportion approaches rely on heuristic thresholds to characterise differences in barcode abundance between samples, which can result in subjective and unreproducible interpretations.

As such, there remains a lack of rigorously validated and robust statistical testing models specifically designed for population based (bulk) barcode count data that can adequately address barcode variability between conditions. A study associated with the DEBRA approach has explored adapting the *DESeq*, *DESeq2*, and *edgeR* models for barcode count data, however, it lacks thorough validation and comprehensive assessment of the performance of these models.[29–31]

This paper presents a Bioconductor package, *barbieQ*, that provides barcode normalization, data-driven filtering, rich visualisations and statistical testing that accommodates the multifactorial designs that are common in the lineage tracing field, enabling reproducible analysis.[32] The statistical tests in *barbieQ* build upon a linear regression framework to account for barcode abundance variation arising from various sources and can robustly detect changes in both barcode abundance and barcode presence/absence between groups of samples. We have used four publicly available datasets to demonstrate and validate *barbieQ*. We found that *barbieQ* controlled false discovery rate using both real and semi-simulated datasets and have demonstrated the ability to detect real barcode differences in a public dataset.

Materials and Methods

Software development and implementation

The *barbieQ* package was developed using R 4.5.0 and Bioconductor version 3.21. We leveraged established libraries ensuring high performance and interoperability within the R Bioconductor ecosystem, such as

- *dplyr* version 1.1.4 for data manipulation, *tydyr* version 1.3.1 for numerical calculations, *SummarizedExperiment* version 1.37.0 and *S4Vectors* version 0.45.4 for data structure.

- *ggplot2* version 3.5.1 and *ComplexHeatmap* version 2.23.0 for drawing plots.
- and *stats* version 4.5.0, *limma* version 3.63.11 and *logistif* version 1.26.0 for statistical modelling.

The package was structured based on the *SummarizedExperiment* S4 class, allowing data manipulation and processing directly within the data structure. Processed results are saved to back to the same structure, extended with additional features to seamlessly interoperate with upstream barcode alignment tools that also use *SummarizedExperiment* or *DGEList* from *edgeR* for data handling.

Data description

Several datasets, briefly described below, were used to demonstrate preprocessing and evaluate the statistical tests implemented in *barbieQ* (see details below). The datasets were obtained from publications introducing tools such as *barcodetrackR*, *cellDestiny*, *bartools* and *DEBRA*. In all cases, the raw barcode count data were downloaded following sequence-level QC and barcode library alignment that had already completed by the original authors.

Monkey HSPC Data

A subset of publicly available data from a study on monkey hematopoietic stem and progenitor cell (HSPC) clonal expansion *in vivo* using barcoding technique.[33] The monkey HSPC data were initially analysed using the *barcodetrackR* package and made available through the compatible barcodetrackRData repository on GitHub (https://github.com/dunbarlabNIH/barcodetrackRData/tree/main/datasets/WuC_etal).[25] Herein, we used the samples from the monkey "ZG66", which has also been included as example data in the *barbieQ* R package.

Briefly, unique barcodes were initially integrated into the DNA of HSPCs and subsequently passed to progeny cells. At a series of time points, progeny cells were collected from blood and sorted into various cell types as samples, including T cells, B cells, granulocytes (Gr), and natural killer (NK) cells for $NK^{CD56+CD16-}$ and $NK^{CD56-CD16+}$ subtypes. Barcode counts within different cell types were used to interpret the patterns of HSPC differentiation. Specifically, the original study endeavoured to find barcodes (clones) associated with $NK^{CD56-CD16+}$ samples showing higher abundance compared to other cell type samples. The barcode count matrix contains 16,603 barcodes, across 53 samples, under the described conditions.

AML Data

Publicly available data from a study on acute myeloid leukaemia (AML) cell clones, investigating the heterogeneity in their response to various therapeutic drugs *in vitro*, which was originally analysed and introduced with *bartools*.[21] Briefly, AML cells were barcoded at the DNA level using the SPLINTR system, recognized as individual clones, and their population was expanded while exposed to different drugs (Arac, IBET), at various doses and DMSO as a negative control.[4] Samples were collected from each treatment condition at a series of time points. This barcode count matrix contains 1,811 barcodes, across 41 samples, under the described conditions.

Mouse HSPC data

Public data from a mouse hematopoietic stem and progenitor cell (HSPC) clonal tracking study, originally analysed and introduced in the *barcodetrackR* publication.[25] In this study, mouse HSPC clones were barcoded at the DNA level, and distinct sets of starting clones were transplanted into individual mice (n = 8). For each mouse, progeny cells were collected from peripheral blood at multiple time points, as well as from various tissues after sacrifice. From each collection, cells were sorted into different cell types forming individual samples, with unsorted cells also retained. The dataset contains over 10,135 barcodes across 8 mice, with approximately 200 samples under the described conditions.

Mixture Data

Dataset generated to simulate both true and null changes in barcode abundance by mixing cells from two barcoded samples.[29] Cells from cell lines were split into two pools, Pool1 and Pool2, then incorporated unique clonal tracking barcodes in their DNA, ensuring non-overlapping barcodes between the two pools. Cells in each

pool were counted and mixed in an equal ratio to produce a mixed pool. Twelve baseline samples were sampled from the mixed pool at various sizes. Twenty-four perturbed samples were generated by sampling a certain number of cells from the mixed pool and adding a certain ratio of cells from Pool1, with 2 replicates in each case. This dataset contains 3998 barcodes, across 36 samples as described, as well as two reference samples representing the unmixed Pool1 and Pool2 barcode counts.

Results

Package overview

The *barbieQ* package workflow consists of data preprocessing and statistical testing, supported by corresponding visualisations. Preprocessing loads in the barcode counts and produces a "*barbieQ*" object, preparing data to be more efficiently and accurately interpreted. Statistical testing of the preprocessed "*barbieQ*" object identifies barcodes with significant changes between sample groups in either differential occurrence or differential proportion. These results can be visualised using dedicated functions. (Figure 1)

Preprocessing starts by "createBarbieQ" reading in a *matrix* of the barcode counts in each sample, optionally including a *data.frame* indicating sample conditions, and initializes a "*barbieQ*" object that is updated throughout the workflow. The "*barbieQ*" object is built upon the *SummarizedExperiment* structure in *R*, consisting of assays storing raw count and processed matrices, barcode-oriented metadata, and sample-oriented metadata. Based on the imported barcode count matrix, the "createBarbieQ" function calculates the barcode count per million (CPM), proportion and ranking per sample, as well as the presence/absence (occurrence), and saves the calculation for each metric as a separate matrix in assays. The definitions of those metrics are listed in Table 2. The function also calculates the diversity index, number of detected barcodes (n_barcode) and number of total barcode counts (n_count) for each sample and saves those sample features as part of the sample data (column data), along with any imported sample experimental conditions. Finally, the function initiates a barcode data (row data) holding space for barcode features derived in subsequent analyses.

"clusterCorrelatingBarcodes" and "collapseBarcodes" can then be used to detect and collapse barcodes co-occurring in the same cell, where "plotBarcodePairCorrelation" can be used for inspecting correlation between every pair of barcodes. Next, "tagTopBarcodes" and "subsetBarcodes" tag barcodes as top/bottom contributors and filter out low-count/low-contribution barcodes, where "plotBarcodeProportion", "plotBarcodePerato" and "plotBarcodeSankey" can be used for inspecting each barcode's contribution before and after filtering. Each step stores the results in the "*barbieQ*" object (Table 2), and recalculates assays such as CPM, proportion, rank, and occurrence when barcodes are collapsed or filtered.

The preprocessed "*barbieQ*" object, generated by collapsing barcodes within the same cell and removing low-abundance barcodes, provides a clean set of barcodes for statistical testing and allows accurate estimation of barcode proportions.

Table 2. *barbieQ* object structure and components

Type	Component	Definition	Initiated by Function
Assay	Count	Barcode count per sample	createBarbieQ
	CPM	Barcode count / total barcode count per sample * 1M	createBarbieQ
	Proportion	Barcode count / total barcode count per sample	createBarbieQ
	Occurrence	Presence/absence: barcode CPM >=1 or CPM <1	createBarbieQ
	Rank	Rank of barcode count per sample	createBarbieQ
	isTop	Top/bottom: barcode being main contributor or low count per sample	tagTopBarcodes
Sample data (*columnData*)	SampleMetadata	Sample experimental conditions (factors)	createBarbieQ
	N_barcode	Number of detected barcodes (total occurrence) per sample	createBarbieQ
	N_count	Number of total barcode counts (library size) per sample	createBarbieQ
	Diversity index	Sample Shannon diversity index	createBarbieQ
	Cluster	Barcode cluster of high correlation	clusterCorrelatingBarcodes

Barcode data (*rowData*)	isTop	Barcode determined as top/bottom considering all samples	tagTopBarcodes
	Significance	Significance of barcode abundance changes between sample groups	testBarcodeSignif

For statistical testing, "testBarcodeSignif" is applied to a "*barbieQ*" object and requires a user-defined design matrix indicating the sample conditions and other experimental factors, as well as user-defined contrasts that specify which sample groups to compare. Then users can choose to test either barcode differential occurrence or differential proportions between the specified sample groups. The results are stored in barcode data in the "*barbieQ*" object (Table 2) and can be visualised by applying dedicated functions to the "*barbieQ*" object, such as "plotBarcodeMA", "plotBarcodePValue", "plotSignifBarcodeProportion", and "plotSignifBarcodeHeatmap".

Additional visualizations, such as "plotSamplePairCorrelation", "plotBarcodeHeatmap" and barcode principal component analysis (PCA), which use barcode CPM, are available before or after statistical testing, to inspect sample heterogeneity and barcode variability.

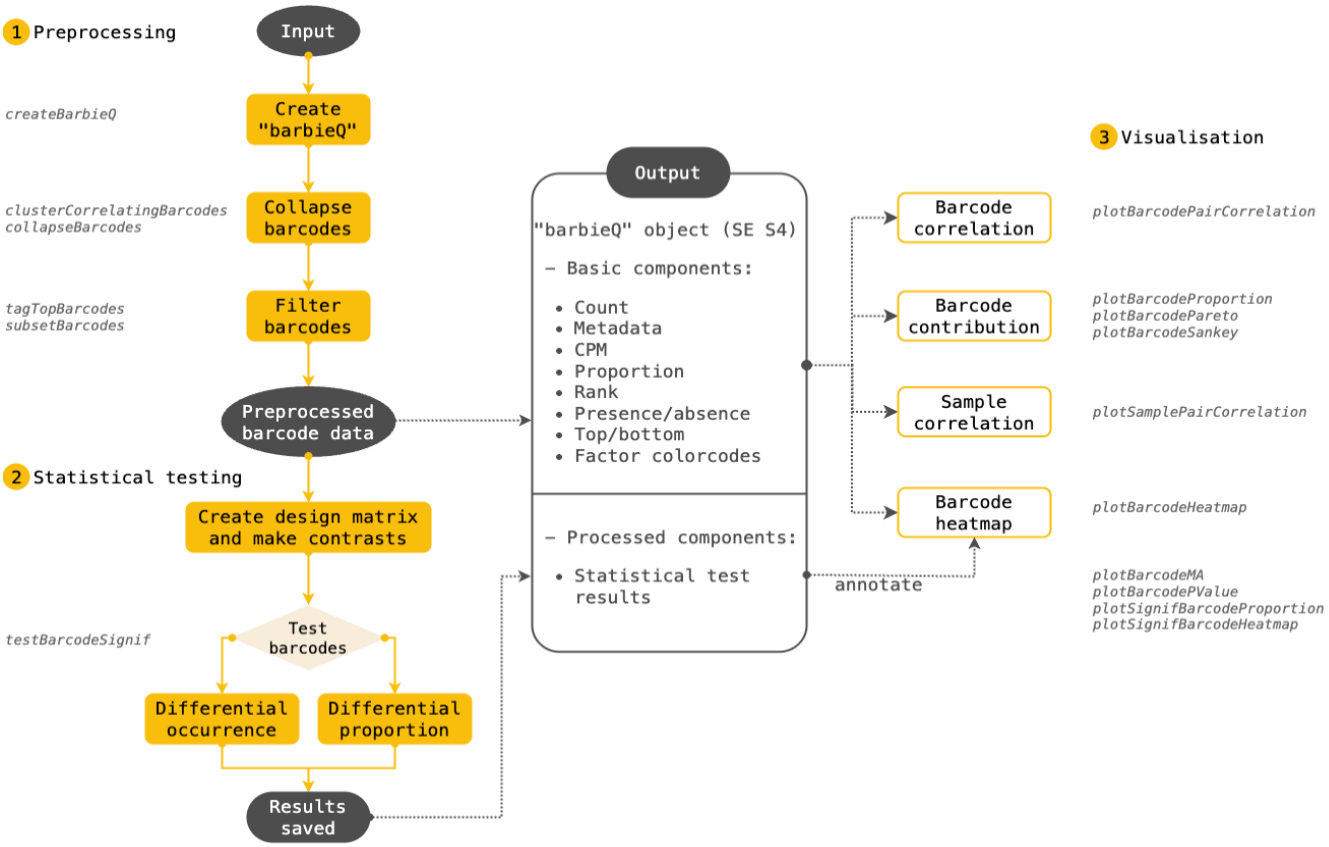

Figure 1. Workflow of *barbieQ* package. *barbieQ* consists of three modules: preprocessing, statistical testing, and visualisation. 1) Preprocessing starts by creating a "*barbieQ*" object, and updates it by collapsing highly correlated barcodes, and filtering out lowly abundant barcodes, ensuring an accurate and clean set of barcodes for subsequent interpretation and visualization. Dedicated functions for each step are listed. 2) Statistical testing applies to the preprocessed "*barbieQ*" object with choices for differential occurrence or differential proportion test, and stores the results in the object, which can be further visualised. 3) Dedicated visualisations are available at any stage.

Barcode Count Preprocessing

We demonstrate the effectiveness of the implemented preprocessing functions by cleaning up an example dataset, the monkey HSPC data, which contains 53 samples.

Collapsing correlating barcodes

The "clusterCorrelatingBarcodes" function is used to identify clusters of highly correlated barcodes, which are likely barcodes that have been technically integrated into the same initial cell and, therefore, co-exist in the progeny cells of the samples. This phenomenon has been observed previously and, if not properly accounted

for, leads to overcounting of cells and distortion of proportions that assumes a one-to-one mapping of barcodes to cells.[21] To address the issue, this function calculates the Pearson correlation coefficient (*r*) of the proportions of every pair of barcodes and classifies highly correlated pairs based on thresholds for correlation coefficient and average CPM. Highly correlated pairs are grouped into clusters of barcodes, each potentially containing two or more barcodes. Barcodes within a cluster are considered to co-exist in one clone and should be merged. "collapseBarcodes" selects the representative barcode as the one with the highest mean CPM for merging to maintain a consistent barcode-clone association.

We applied this approach to the monkey HSPC data. For efficiency, low-count barcodes were filtered out beforehand using the methods described in the next section. Barcode pairs with a correlation coefficient greater than 0.95 and an average mean CPM exceeding 2^{10} were classified as paired barcodes belonging to a cluster (Figure 2A). In total, 24 barcodes formed 9 clusters (Figure 2B), each containing 2~6 barcodes (Figure 2C). The mean CPM of the barcodes within a cluster varied (Figure 2D). These barcode clusters were subsequently collapsed for downstream analysis. These findings are consistent with a published study, and an unpublished dataset we analysed, where repeated barcodes were validated through single-cell RNA sequencing of the initial cells, confirming the co-existence and co-expression of the repeated DNA barcodes (Supplementary Figure 1A).[21] Preprocessing results for the three additional datasets (AML data, mouse HSPC, and mixture data) can be found in Supplementary Figure 1.

Barcode filtering

The "tagTopBarcodes" function uses a data-driven approach to classifies barcodes as "top" and "bottom" contributors, which guides filtering to retain "top" barcodes that contribute substantially to the total counts while removing "bottom" barcodes that exhibit minimum contributions. Barcode proportions are calculated for each sample and ranked from highest to lowest. A cumulative proportion is then calculated at each rank, starting from the most abundant barcode. Barcodes ranked ahead of the cumulative proportion cutoff at an empirical threshold (default = 0.99) are tagged as "top" in a sample, indicating they are the most abundant and contribute the most to the sample. Across the dataset, barcodes being tagged as "top" in at least *n* samples (*n* being the minimum sample group size) are classified as "top" and subsequently retained across the whole dataset. This approach ensures that consistently low-abundance barcodes are filtered out while preserving those present in at least one biological replicate group.

We applied this approach to the monkey HSPC dataset, using the default proportion cutoff of 0.99 and a minimum group of n = 6; this was chosen because the 42 samples feature 5 groups of cell types with a minimum group size of 6. This resulted in the classification of 1378 "top" barcodes out of a total of 16579. The mean proportion of each barcode was visualised with the *barbieQ* functions "plotBarcodeProportion" and is shown in Figure 2E. These "top" barcodes represent 8% of all barcodes accounting for 93% of total barcode counts in the dataset as shown in Figure 2F using the "plotBarcodeSankey" visualisation. By filtering out consistently low abundance "bottom" barcodes, we retained a focused, clean, and interpretable set of "top" barcodes as shown in Figure 2G, H.

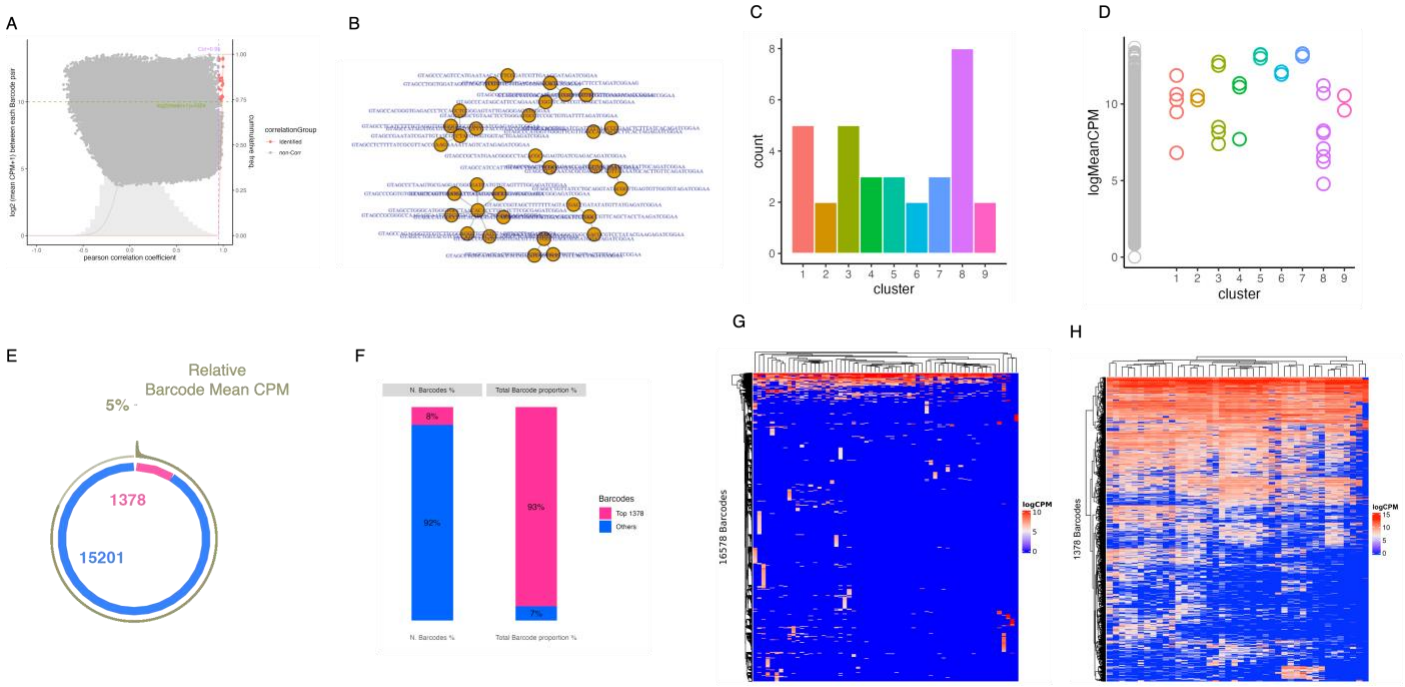

Figure 2. Preprocessing results on monkey HSPC data. (**A**) Log mean CPM and Pearson correlation coefficients for each pair of barcodes out of 1667 barcodes using the "plotBarcodePairCorrelation" function; (**B**) Cluster of highly correlated barcodes visualised using the "clusterCorrelatingBarcodes" function; (**C**) Number of barcodes in each cluster; (**D**) Log mean CPM of each barcode within its cluster; (**E**) Proportion of each barcode's mean CPM out of all barcodes (outside circle), color-coded by "top" or "bottom" contributors as tagged by the filtering strategy (inside circle), using the "plotBarcodePareto" function; (**F**) Fraction of "top" or "bottom" contributor barcodes and their cumulative mean proportion, using the "plotBarcodeSankey" function; (**G, H**) Heatmap of barcodes on logCPM before and after filtering as visualised using the "plotBarcodeHeatmap" function.

Performance of *barbieQ* statistical testing

To assess differences in barcode counts between sample groups, that represent clonal outcomes, *barbieQ* can interpret barcode changes in two ways: differential proportion and differential occurrence. Differential proportion identifies barcodes with significantly different proportions between the samples of two conditions, whilst differential occurrence identifies barcodes that are more frequently present in samples of one condition compared to another.

Both statistical tests are based on a generalised linear regression model that assesses each barcode individually, while accounting for barcode variability between samples groups. A key feature of regression models is their ability to deal with complex experimental designs using a design matrix that takes account of multiple factors and thereby estimates the variance after fitting each factor. This enables the tests to determine whether the factor of interest significantly changes barcode occurrence or proportion, while accounting for the effects of other confounding factors. Differential proportion relies on the "*limma*" linear regression framework.[34,35] It models barcode proportions after applying an arcsine square root (asin-sqrt) transformation to stabilize the variance across the proportion range to satisfy the assumptions of the linear regression. The logit transformation is an alternative transformation available in *barbieQ*.

Differential occurrence uses a logistic regression model combined with Firth's penalised likelihood to reduce bias in the maximum likelihood estimates of regression coefficients. This correction addresses inflated standard error that can arise in extreme cases of perfect separation, which are more likely to occur in small sample size cases (e.g., n = 3), not uncommon in clonal tracking studies. The model takes barcode presence or absence in each sample as a binary value, determined by CPM ≥ threshold (default to 1) for presence, and CPM < threshold for absence.

The function "testBarcodeSignif" allows users to specify the statistical test and define the factor of interest through a design matrix with the desired contrast. The results of the test are saved to the *rowData* of the *barbieQ* object, aligned with the corresponding barcodes. The results contain key statistics such as group-wise means

and differences in proportion or odds ratios in occurrence, along with *P*-values adjusted for multiple testing using the Benjamin-Hochberg method by default.[36] This "*barbieQ*" object can then be passed to visualisation approaches for inspection as outlined in Figure 1: "plotSignifBarcodeProportion", "plotSignifBarcodeHeatmap", "plotBarcodeMA" and "plotBarcodePValue".

Type I error control under null scenario using real-world data

To evaluate the type I error rates of the two statistical tests implemented in *barbieQ*, we applied these tests to a variety of datasets. Within each dataset, we selected samples with homogeneous conditions, or those not designed to show biologically meaningful differences. These samples were treated as replicates for random sampling and used for type I error evaluation by artificially defining two groups of samples from the set. More specifically, from the set of samples, we generated 100 simulated null datasets by randomly assigning samples into two groups without replacement and then applied the differential occurrence test and differential proportion test, using the "testBarcodeSignif" function, to assess the number of barcodes with significant (*P*-value < 0.05) change between the two groups. We also varied the number of samples per group to determine the effect of group size on Type I error rate (n = 3, 4, ..., up to half of the total number of the null samples).

Here we demonstrate the type I error control using a subset of the Mixture dataset (Figure 3A). Cells in Pool1 and Pool2 were counted and mixed in an equal ratio to produce a mixed pool. Twelve baseline samples were sampled from the mixed pool at various total cell numbers (n = 20, 40, 80, 160, 330, and 660 thousand, each with two replicates). Difference within these baseline samples represent samples under the null hypothesis – i.e. there should be no differences between relative barcode abundance, except for the technical effects of total number of cells (number of sequenced barcodes).

Filtering from the total 3998 barcodes, we retained 2883 barcodes, with mean proportion ranging from 3×10^{-6} to 0.03. (Supplementary Figure1) Notably, barcode proportion used for type I error analysis was further generated based on adding 1 to the count. To investigate how transformation influences the differential proportion test, we applied the test after different data transformations: arcsine square root (asin-sqrt), logit and no transformation. Figure 3C summarizes the Type I error rates for 100 randomly selected groups of samples, for different group sizes and different transformations. We found most simulations controlled the type 1 error with the median Type I error rate across 100 simulations under 0.05 for all tests. We also observed that the samples in simulations with high Type I error rate are consistent between the three tests, (Supplementary Figure2). Typically, these outlier simulations tend to have randomly grouped the samples by library size (high vs. low).

We found that asin-sqrt transformation holds its size better than logit transformation for the differential proportion test, (Figure 3C). To explore this further, we binned the barcodes into 10 quantiles based on their mean proportion and compared the faction of barcodes with significant *P*-values within the bins obtained from those tests (Figure 3D). Compared with asin-sqrt transformation, logit tends to yield more significant *P*-Values when barcode mean proportion fell in the middle range, and less significant barcodes *P*-Values when barcode mean proportion was higher. (Figure 3D) Notably, nearly half barcodes fell in (10^{-5} ~10^{-4}], explaining the overall high FPR of logit transformation in Figure 3C. Asin-sqrt and no transformation showed relatively consistent performance in FPR across all barcode proportion range and produced more type 1 errors at higher proportions. It should be noted that this data has a large number of barcodes compared with some other data sets resulting in relatively small proportions.

We also evaluated Type I error rates using subsets of samples from the mouse HSPC dataset (Supplementary Figure3), where samples collected from the same mouse (C21, C22 and C32) were treated as null samples, despite the differences in time points, tissues, and cell types. All tests maintained the Type I error rates under 0.05 in the null simulations. A similar trend in *P*-values between transformations was also observed in these data.

Power to detect true differences in barcode proportions using the Mixture dataset

Next, we evaluated the power of the differential proportion test to detect true differences in barcode proportions between sample groups using a subset of the prepossessed Mixture dataset. Twenty-four perturbed samples were generated by introducing additional Pool1 cells into the baseline level: sampling a certain number of cells

from the mixed pool (n= 20, 40, 80 or 160 thousand) and adding a certain ratio of cells from Pool1 (α = 0.18, 0.27, or 0.35), with 2 replicates in each case. Specifically, (n × α) cells from Pool1 were added to the n cells from the mixed pool to generate a perturbed sample. The previously discussed 12 baseline samples were considered as perturbation level α = 0. (Figure 3A) This design leads to a known difference in barcode proportions (for all barcodes) between samples with different α.

In total, 3998 unique barcodes were observed in this dataset, with barcode proportion ranging from a maximum of 0.057, a mean of approximately 2.5×10^{-4} and a median of 1.4×10^{-5}. We preprocessed the data as outlined in Figure 1, removing low count barcodes considering all 36 samples and retained 2883 barcodes.

The mixture pool contains a mix of cells with barcodes from Pool1 and Pool2, for which the barcodes didn't overlap. Let x_n be the proportion of barcode, n = (1, ..., 2883).

$$X_{Pool1} = \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad X_{Pool2} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_{m+1} \\ \vdots \\ x_n \end{bmatrix}$$

A baseline sample theoretically consisted of an equal mixture of total barcode numbers of Pool1 and Pool2, leading to barcode proportions being half of the reference samples.

$$X_0 = \frac{1}{2}(X_{Pool1} + X_{Pool2})$$

A sample with a perturbation level of α theoretically leads to a known change in proportion for the barcodes from both Pool and Pool2.

$$X_\alpha = \frac{\frac{1}{2}(1+\alpha)}{(1+\alpha)} X_{Pool1} + \frac{\frac{1}{2}}{(1+\alpha)} X_{Pool2}, \qquad \alpha = 0.18, 0.27, 0.35$$

The designed difference in barcode proportions between perturbed samples and baseline samples can therefore be calculated for each transformation.

$$\Delta = f(X_\alpha) - f(X_0), \qquad where\ f_{asin-sqrt}(x) = \sin^{-1}(\sqrt{x}), f_{logit}(x) = log\frac{x}{1-x}, f_{no\ trans}(x) = x, \qquad x \in (0,1)$$

Therefore, for this data we modelled the 36 samples using a design matrix that set factors for each perturbation level (α = 0, 0.18, 0.27, and 0.35), with additional factors taking into account the total number of barcodes in the equal mixture (n= 20, 40, 80, 160, 330, and 660 thousand) before adding Pool1 barcodes. We then applied the "testBarcodeSignif" function and tested barcode differential proportions for three contrasts: $group_{\alpha=0.18}$, $group_{\alpha=0.27}$, and $group_{\alpha=0.35}$ v.s. $group_{\alpha=0}$, respectively, each with asin-sqrt, logit and no transformation. The estimated coefficients of the contrasts obtained from the "testBarcodeSignif" function give the estimated differences between sample groups for each barcode. Figure 3E plots the designed and estimated differences for each contrast with different transformation methods. Both asin-sqrt and no transformation accurately estimated the barcode proportion change, while logit tended to over-estimate the change, although the designed differences for logit were sensitive to the non-zero small count in the reference samples.

We then evaluated the sensitivity of the differential proportion tests with three transformations using *P*-values of barcodes obtained from those contrasts. All barcodes were designed to change their proportions within those contrasts, either increase or decrease. The barcodes were binned into 10 quantiles by mean proportion and then the fraction of barcodes with *P*-values < 0.05 within each bin was calculated for each contrast with different transformations. (Figure 3F) Asin-sqrt and no transformation showed similar sensitivity across all ranges, both showing higher sensitivity than logit when barcode mean proportion was greater than 10^{-4}.

Finally, we inspected the distribution of residuals obtained from fitting barcode proportion by sample perturbation levels using the linear regression model implemented in the "testBarcodeSignif" function. (Figure G, H)

Although asin-sqrt and no transformation showed similar performance in maintaining type I error rate and sensitivity, the residuals from asin-sqrt transformed proportion better meet the assumptions of normality.

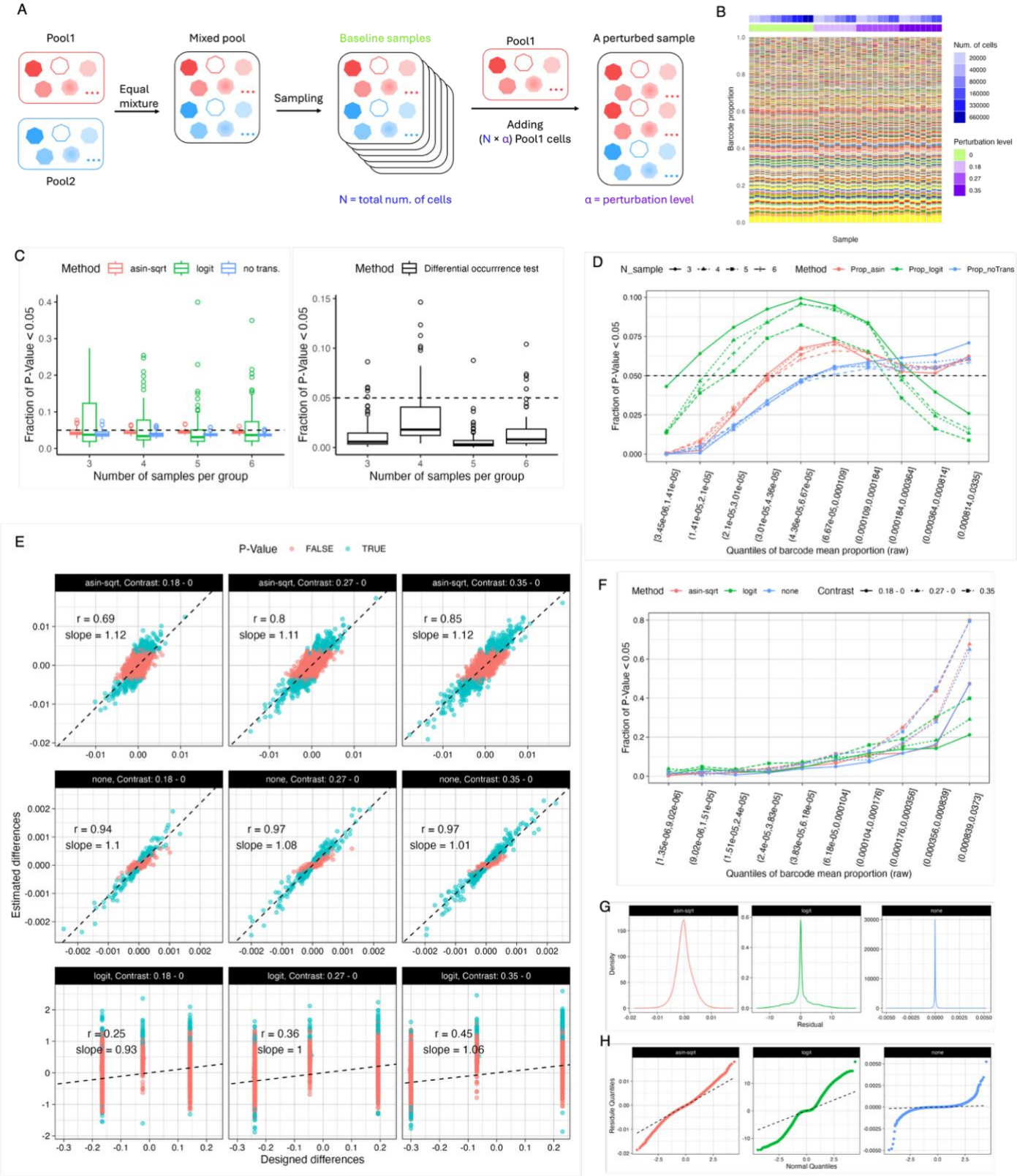

Figure 3. Performance of *barbieQ* statistical testing. (**A**) Experimental design showing the generation of baseline samples and perturbed samples in the Mixture dataset; (**B**) Barcode proportion per samples in the Mixture dataset, with each barcode represented by a unique colour; Samples are annotated by total number of cells and perturbation levels; (**C**) Type I error rates of differential proportion test with three transformation methods (left) and differential occurrence test (right), evaluated by testing simulated groups within the baseline samples. The faction of barcodes with raw *P*-value < 0.05 was averaged across 100 simulations at different group sizes. The three differential proportion tests added 1 to the raw count before calculating proportions to address under sampling effects caused by low library sizes in some samples; (**D**) Raw *P*-values in differential proportion tests in Figure 3**C** binned into 10 quantiles by barcode mean proportion where the faction of barcodes with raw *P*-value < 0.05 in each bin for different transformation methods and group sizes was calculated; (**E**) Estimated differences in barcode proportions between perturbed samples (perturbation level = 0.18, 0.27, 0.25) and

baseline samples (perturbation level = 0), obtained from the estimated coefficients of the contrast using differential proportion test with three transformation methods; Designed differences in barcode proportion for each contrast under different transformations, calculated based on original barcode proportion in Pool1 and Pool2 samples and experimental design in Figure 3**A**. (**F**) Raw *P*-values in differential proportion tests in Figure 3**E** binned into 10 quantiles of mean barcode proportion where the faction of barcodes with raw *P*-value < 0.05 in each bin for different transformation methods and contrasts was calculated; (**G**) The distribution of residuals and QQ plots obtained from linear regression modelling implemented in the differential proportion test with three transformation methods.

Case study

Reanalysis of monkey HSPC data

Two important features of the *barbieQ* package are its data-driven filtering strategy and robust statistical tests. To demonstrate how *barbieQ* advances upon previous approaches in the clonal tracking and lineage tracing fields, we applied the *barbieQ* analytical workflow to a real-world dataset and compared the results with those reported in the original publication. The example data is a subset of 30 samples from the monkey HSPC dataset, collected across 6 time points and sorted into 5 different cell types (Figure 4A). Figure 4B shows the barcode proportion per sample. In total, 16,603 unique barcodes were observed in this dataset, with barcode proportion ranging from a maximum of 0.14, a mean of approximately $6x10^{-5}$ and a median of 0. These barcode proportions are variable between cell types but relatively stable between time points within the same cell type.

We preprocessed the raw barcode count data following the steps outlined in Figure 2, including aggregating highly correlated barcodes and filtering low-count barcodes. Figure 4C shows a Venn diagram comparing barcodes retained by the *barbieQ* filtering strategy and commonly used *top10-* or *top20- barcodes-per-sample* approaches (as used in the original publication). *barbieQ* retained a larger number of barcodes including those retained by the *topN*-based approaches. Figure 4D shows Sankey plots for the faction of retained barcodes and their total contribution to the entire dataset under each strategy. *barbieQ* retained 8% of barcodes, accounting for 93% of the total counts while *top10-per-sample* retained 0.5% of barcodes contributing 43% of total counts, and *top20-per-sample* retained 1% of barcodes contributing 51% of total counts. Figure 4E shows a heatmap of barcode logCPM, annotating barcodes retained by both *barbieQ* and *top10-per-sample* strategies or exclusively by one of them. Notably, barcodes exclusively retained by *barbieQ* still show informative logCPM across samples. An additional advantage of *barbieQ* is avoiding arbitrarily selecting a *topN* threshold for every new dataset, making the filtering process less subjective.

Next, we applied the statistical testing methods implemented in *barbieQ* by reproducing the comparisons in the original publication using the exact same sets of barcodes. The first analysis includes 81 barcodes that were selected by the *top10-per-sample* filtering strategy out of the 30 samples (Figure 4A, B). We used *barbieQ* to test barcode differences between $NK^{CD56-CD16+}$ samples and other cell types (T, B, Gr and $NK^{CD56+CD16-}$), treating each individual time point as a factor. We modelled the design matrix that sets the 5 cell types as 5 factors and 6 time points as 6 additional factors and created t contrast as $NK^{CD56-CD16+}$ - (T + B + Gr + $NK^{CD56+CD16-}$) / 4. We applied the differential proportion test, using the "testBarcodeSignif" function, based on asin-sqrt transformation to assess changes in barcode proportion between $NK^{CD56-CD16+}$ samples and others. We identified 23 barcodes with significantly increased proportion in $NK^{CD56-CD16+}$ samples, including 6 novel barcodes and 17 overlapping with those selected by the original publication, which used a heuristic cutoff for barcode proportion ratios between $NK^{CD56-CD16+}$ and other cell types at each time point respectively (Figure 4F). We inspected the log fold change (LFC) in barcode proportions ($NK^{CD56-CD16+}$ / other cell types) at different time points and observed that the 6 novel barcodes showed robust LFC (Figure 4F).

In addition, we applied the same test to a larger set of barcodes retained using the *barbieQ* filtering strategy, including 1,029 barcodes that contribute 99% to the 30 samples. We identified three times more barcodes showing significantly increased proportion in $NK^{CD56-CD16+\ cells}$ and collectively contributing more to the $NK^{CD56-CD16+}$ samples (Figure 4G, H). We inspected the LFC of the significant barcodes obtained based on either filtering strategy. (Figure 4J) Using the *barbieQ* filtering strategy allowed us to keep more barcodes with robust LFC and relatively high abundance.

The second analysis used the subset of 8 $NK^{CD56-CD16+}$ samples from the monkey HSPC dataset, collected at various time points. We investigated the 39 barcodes retained after filtering by the *top10-per-sample* filtering

strategy to match the set of barcodes in the original publication. We then used the "testBarcodeSignif" function to test barcodes that change over time within NK$^{CD56-CD16+}$ samples. Specifically, each sample was given a numeric factor for the number of months post transplantation. We tested each barcode for significant change over time. We identified 17 barcodes with significantly increased and 8 with decreased proportions over time (Figure 4I). The *barbieQ* testing results robustly categorized barcode kinetics, compared to selecting barcodes by visual inspection on heatmap dendrograms as used in the original publication.

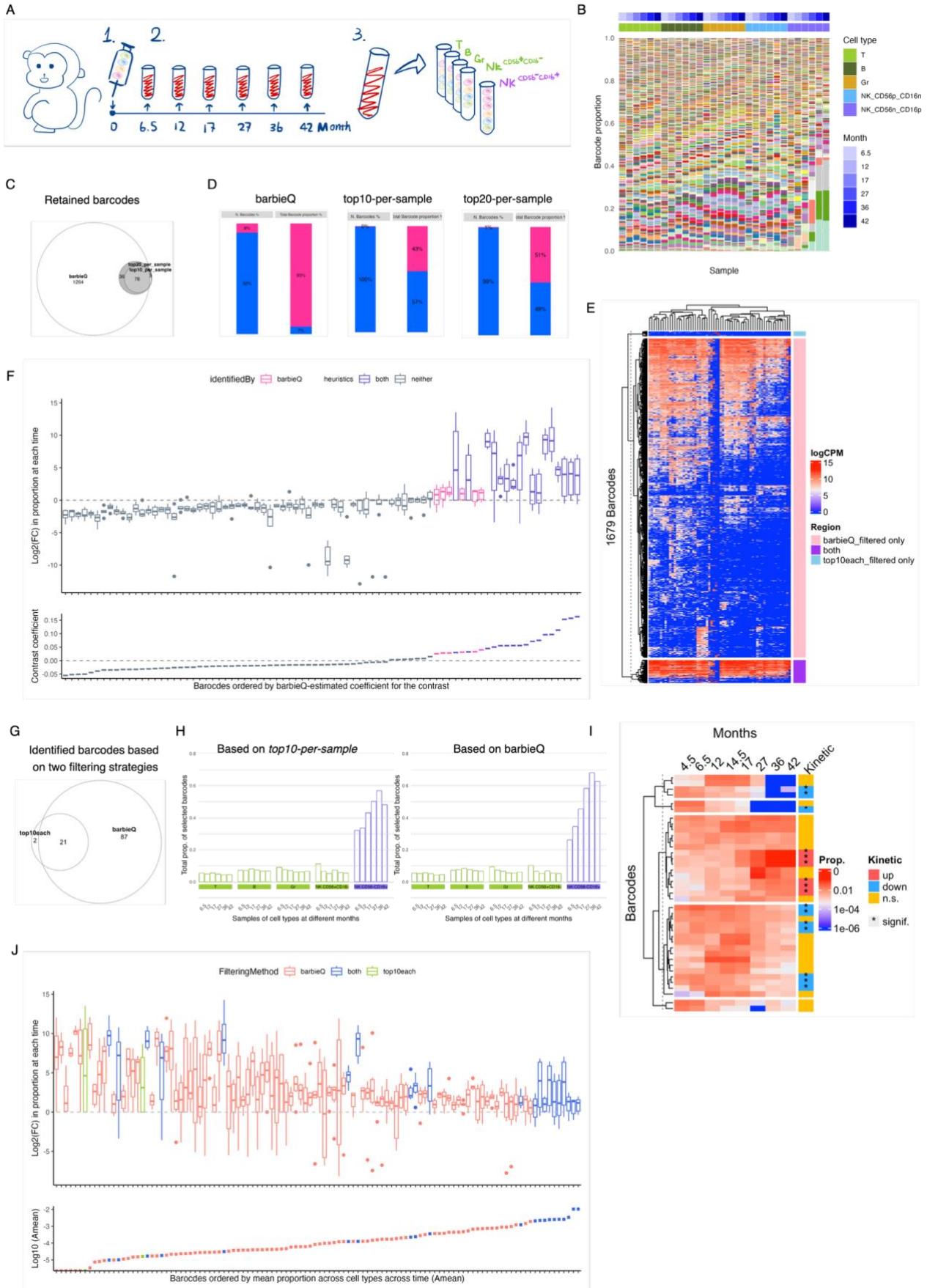

Figure 4 Applying *barbieQ* to the monkey HSPC dataset. (**A**) Experimental design for a subset of the monkey HSPC dataset, where 30 samples contained 5 cell types sharing 6 time points; (**B**) Barcode proportion per samples in the subset Monkey HSPC dataset, with each barcode represented by a unique colour; Samples are annotated by cell types and collected months; (**C**) Euler diagram showing the overlap in retained barcodes using *barbieQ* filtering, *top10-per-sample* and *top20-per-sample* strategies; (**D**) Faction of retained barcodes and their total relative mean proportion to the whole dataset. Filtering strategies from left to right: *barbieQ*, *top10-per-sample*, and *top20-per-sample*; (**E**) Heatmap of retained barcodes by either *barbieQ* or *top10-per-sample.* Right annotations indicate the overlap between the two approaches; (**F**) Box plot of the LFC for each barcode. A dot refers to the LFC in proportion between to $NK^{CD56-CD16+}$ and other cell types at a given time points (6 time points in total). Barcode were ordered by *barbieQ*-estimated coefficient for the contrast, from low to high. Barcodes were categorized into identified as $NK^{CD56-CD16+}$-associated by *barbieQ*, heuristic LFC, both and neither method. (**G**) Eular diagram showing the overlap in identified $NK^{CD56-CD16+}$-associated barcodes using *barbieQ* "testBarcodeSignif" based on *barbieQ* filtering and *top10-per-sample* strategies. (**H**) Total barcode proportion per sample for identified $NK^{CD56-CD16+}$-associated barcodes based on *top10-per-sample* (left) and *barbieQ* filtering strategy (right), respectively. (**J**) Box plot of the LFC for each barcode. A dot refers to the LFC in proportion between to $NK^{CD56-CD16+}$ and other cell types at a given time points (6 time points in total). Barcodes were ordered by mean proportion across cell types and time points, from low to high. Barcodes were categorized into identified as $NK^{CD56-CD16+}$-associated by *barbieQ*, based on the set of *barbieQ* filtering and top10-per-sample, and both strategies. (**I**) Heatmap of 39 barcodes retained from 8 $NK^{CD56-CD16+}$ samples using *barbieQ* filtering. Samples were order over time. Right annotations indicate barcodes being identified with significantly increasing, decreasing kinetics or not-significant (n.s.) over time. Barcodes were grouped into 5 patterns using the left dendrogram.

Discussion

Lineage tracing experiments provide powerful insights into cellular development and evolution, but the complexity of the resulting data presents significant analytical challenges. Current approaches, typically based on ad hoc filtering or simple comparisons, may fail to account for the inherent complexity of lineage tracing datasets, potentially leaving important insights unexplored. Here, we presented *barbieQ*, a new Bioconductor package tailored to the analysis of lineage tracing barcode count data that provides an integrated platform for quality control, filtering, visualisation and statistical testing. Built upon the *SummarizedExperiment* infrastructure, it ensures compatibility with existing workflows while introducing two tailored statistical tests for identifying differential barcode proportion and differential barcode occurrence. These tests enable the robust identification of barcode differences between sample groups, whilst comprehensively accounting for variability resulting from multiple technical and biological conditions. *barbieQ*'s data-driven approach to preprocessing using the functions clusterCorrelatingBarcodes and tagTopBarcodes, guarantees a clean, yet informative set of barcodes for testing. Additionally, the built-in visualisation functions can be used to inspect barcode features throughout an analysis, ensuring transparency and interpretability at every stage of the workflow.

We demonstrated the *barbieQ*'s preprocessing capabilities by applying the clusterCorrelatingBarcodes function to four barcode datasets: the monkey HSPC, AML data, mouse HSPC, and the mixture data. In the first three, we found highly correlated barcode clusters likely to be the multiple barcodes that were incorporated into the same clones during preparation. Although we have successfully applied this approach in the analysis of an unpublished dataset with truth validated by single cell sequencing, it has obvious limitations in its reliance on heuristic thresholds that may not generalise across different datasets. However, as the incorporation of multiple barcodes into a single cell becomes increasingly unlikely due to advancements in laboratory techniques, even with this limitation, the clusterCorrelatingBarcodes function remains useful for confirming barcode uniqueness. We also applied the tagTopBarcodes function to the four datasets to identify the main contributing barcodes across samples. We showed that our data-driven method effectively filters the data, yet retains the most informative barcodes. We demonstrated the effectiveness of the *barbieQ* filtering strategy in comparison to the commonly used *top-10* (or *20*)-per-sample approach in the analysis of monkey HSPC data, where our approach retained a greater number of potentially relevant barcodes.

We validated the performance of *barbieQ*'s statistical testing using the mixture dataset. The testBarcodeSignif function is implemented with two primary options for statistical testing: differential proportion and differential occurrence. We showed that, in general, both tests hold their size. For the differential proportion test, we investigated two different commonly-used transformations for proportion data: the arcsin square root transformation (asin) and the logit transformation (logit), in order to accommodate the features of barcoding data, which often have hundreds to thousands component barcodes, with extremely low proportions. Through semi-

simulation using a subset of the mixture data, we found that differential occurrence and differential proportion (asin) have a good false positive rate control, while differential proportion (logit) does not. Through analysis of a second subset of the mixture data with designed differences, we found that differential proportion (asin) has more power to identify real changes in barcode proportions. This suggests that in a scenario with such typical distribution of barcode proportion, where most barcode proportions fall in under 10^{-4}, differential proportion (asin) is a better choice than differential proportion (logit). However, in the semi-simulation study, we also found that differential proportion (logit) has a superior false positive rate control in high proportions (10^{-4}~10^{-2}) despite a its poor performance in the low-proportion range (10^{-5}~10^{-4}). This suggests that differential proportion (logit) may be a good alternative in datasets with fewer barcodes (or features) and generally higher proportions, which is usually the case of single cell type proportions. We also applied the *barbieQ* workflow to the analysis of the monkey HSPC data, where samples differed in time points and cell types. We repeated analyses from the original paper using the *barbieQ* differential proportion test and identified a novel set of barcodes with significantly higher proportion in mature NK cells than other cell types. We also robustly identified a set of barcodes showing significantly increasing proportion over time in the mature NK cell cohort, some of which were only described using a heatmap in the orginal paper. This demonstrates that *barbieQ* can recapitulate existing results derived using heuristics and visualisation, whilst robustly and reproducibly identifying additional barcodes of interest. Furthermore, given that *barbieQ*'s differential proportion test is built upon the versatile *limma* framework, its usage can be extended beyond simple group comparisons, such as fitting "splines" to identify more complex barcode kinetics.

In conclusion, *barbieQ* is a powerful Bioconductor package for robustly and reproducibly detecting biologically meaningful changes in lineage tracing barcode count data. The *barbieQ* package is available from Bioconductor.[32]

Reference

(1) Bramlett, C.; Jiang, D.; Nogalska, A.; Eerdeng, J.; Contreras, J.; Lu, R. Clonal Tracking Using Embedded Viral Barcoding and High-Throughput Sequencing. *Nat. Protoc.* **2020**, *15* (4), 1436–1458. https://doi.org/10.1038/s41596-019-0290-z.

(2) Lin, D. S.; Tian, L.; Tomei, S.; Amann-Zalcenstein, D.; Baldwin, T. M.; Weber, T. S.; Schreuder, J.; Stonehouse, O. J.; Rautela, J.; Huntington, N. D.; Taoudi, S.; Ritchie, M. E.; Hodgkin, P. D.; Ng, A. P.; Nutt, S. L.; Naik, S. H. Single-Cell Analyses Reveal the Clonal and Molecular Aetiology of Flt3L-Induced Emergency Dendritic Cell Development. *Nat. Cell Biol.* **2021**, *23* (3), 219–231. https://doi.org/10.1038/s41556-021-00636-7.

(3) Bandler, R. C.; Vitali, I.; Delgado, R. N.; Ho, M. C.; Dvoretskova, E.; Ibarra Molinas, J. S.; Frazel, P. W.; Mohammadkhani, M.; Machold, R.; Maedler, S.; Liddelow, S. A.; Nowakowski, T. J.; Fishell, G.; Mayer, C. Single-Cell Delineation of Lineage and Genetic Identity in the Mouse Brain. *Nature* **2022**, *601* (7893), 404–409. https://doi.org/10.1038/s41586-021-04237-0.

(4) Fennell, K. A.; Vassiliadis, D.; Lam, E. Y. N.; Martelotto, L. G.; Balic, J. J.; Hollizeck, S.; Weber, T. S.; Semple, T.; Wang, Q.; Miles, D. C.; MacPherson, L.; Chan, Y.-C.; Guirguis, A. A.; Kats, L. M.; Wong, E. S.; Dawson, S.-J.; Naik, S. H.; Dawson, M. A. Non-Genetic Determinants of Malignant Clonal Fitness at Single-Cell Resolution. *Nature* **2022**, *601* (7891), 125–131. https://doi.org/10.1038/s41586-021-04206-7.

(5) Weinreb, C.; Rodriguez-Fraticelli, A.; Camargo, F. D.; Klein, A. M. Lineage Tracing on Transcriptional Landscapes Links State to Fate during Differentiation. *Science* **2020**, *367* (6479), eaaw3381. https://doi.org/10.1126/science.aaw3381.

(6) Naik, S. H.; Perié, L.; Swart, E.; Gerlach, C.; van Rooij, N.; de Boer, R. J.; Schumacher, T. N. Diverse and Heritable Lineage Imprinting of Early Haematopoietic Progenitors. *Nature* **2013**, *496* (7444), 229–232. https://doi.org/10.1038/nature12013.

(7) Lin, D. S.; Kan, A.; Gao, J.; Crampin, E. J.; Hodgkin, P. D.; Naik, S. H. DiSNE Movie Visualization and Assessment of Clonal Kinetics Reveal Multiple Trajectories of Dendritic Cell Development. *Cell Rep.* **2018**, *22* (10), 2557–2566. https://doi.org/10.1016/j.celrep.2018.02.046.

(8) Guo, C.; Kong, W.; Kamimoto, K.; Rivera-Gonzalez, G. C.; Yang, X.; Kirita, Y.; Morris, S. A. CellTag Indexing: Genetic Barcode-Based Sample Multiplexing for Single-Cell Genomics. *Genome Biol.* **2019**, *20* (1), 90. https://doi.org/10.1186/s13059-019-1699-y.

(9) Liu, K.; Deng, S.; Ye, C.; Yao, Z.; Wang, J.; Gong, H.; Liu, L.; He, X. Mapping Single-Cell-Resolution Cell Phylogeny Reveals Cell Population Dynamics during Organ Development. *Nat. Methods* **2021**, *18* (12), 1506–1514. https://doi.org/10.1038/s41592-021-01325-x.

(10) Biddy, B. A.; Kong, W.; Kamimoto, K.; Guo, C.; Waye, S. E.; Sun, T.; Morris, S. A. Single-Cell Mapping of Lineage and Identity in Direct Reprogramming. *Nature* **2018**, *564* (7735), 219–224. https://doi.org/10.1038/s41586-018-0744-4.

(11) Deng, L.-H.; Li, M.-Z.; Huang, X.-J.; Zhao, X.-Y. Single-Cell Lineage Tracing Techniques in Hematology: Unraveling the Cellular Narrative. *J. Transl. Med.* **2025**, *23* (1), 270. https://doi.org/10.1186/s12967-025-06318-4.

(12) Weber, T. S.; Biben, C.; Miles, D. C.; Glaser, S. P.; Tomei, S.; Lin, C.-Y.; Kueh, A.; Pal, M.; Zhang, S.; Tam, P. P. L.; Taoudi, S.; Naik, S. H. LoxCode in Vivo Barcoding Reveals Epiblast Clonal Fate Bias to Fetal Organs. *Cell* **2025**, *0* (0). https://doi.org/10.1016/j.cell.2025.04.026.

(13) Putri, G. H.; Pires, N.; Davidson, N. M.; Blyth, C.; Al'Khafaji, A. M.; Goel, S.; Phipson, B. Extraction and Quantification of Lineage-Tracing Barcodes with NextClone and CloneDetective. November 20, 2023. https://doi.org/10.1101/2023.11.19.567755.

(14) Thielecke, L.; Cornils, K.; Glauche, I. genBaRcode: A Comprehensive R-Package for Genetic Barcode Analysis. *Bioinformatics* **2020**, *36* (7), 2189–2194. https://doi.org/10.1093/bioinformatics/btz872.

(15) Serrano, A.; Berthelet, J.; Naik, S. H.; Merino, D. Mastering the Use of Cellular Barcoding to Explore Cancer Heterogeneity. *Nat. Rev. Cancer* **2022**, *22* (11), 609–624. https://doi.org/10.1038/s41568-022-00500-2.

(16) Kong, W.; Biddy, B. A.; Kamimoto, K.; Amrute, J. M.; Butka, E. G.; Morris, S. A. CellTagging: Combinatorial Indexing to Simultaneously Map Lineage and Identity at Single-Cell Resolution. *Nat. Protoc.* **2020**, *15* (3), 750–772. https://doi.org/10.1038/s41596-019-0247-2.

(17) Oren, Y.; Tsabar, M.; Cuoco, M. S.; Amir-Zilberstein, L.; Cabanos, H. F.; Hütter, J.-C.; Hu, B.; Thakore, P. I.; Tabaka, M.; Fulco, C. P.; Colgan, W.; Cuevas, B. M.; Hurvitz, S. A.; Slamon, D. J.; Deik, A.; Pierce, K. A.; Clish, C.; Hata, A. N.; Zaganjor, E.; Lahav, G.; Politi, K.; Brugge, J. S.; Regev, A. Cycling Cancer Persister Cells Arise from Lineages with Distinct Programs. *Nature* **2021**, *596* (7873), 576–582. https://doi.org/10.1038/s41586-021-03796-6.

(18) Emert, B. L.; Cote, C. J.; Torre, E. A.; Dardani, I. P.; Jiang, C. L.; Jain, N.; Shaffer, S. M.; Raj, A. Variability within Rare Cell States Enables Multiple Paths toward Drug Resistance. *Nat. Biotechnol.* **2021**, *39* (7), 865–876. https://doi.org/10.1038/s41587-021-00837-3.

(19) Goyal, Y.; Busch, G. T.; Pillai, M.; Li, J.; Boe, R. H.; Grody, E. I.; Chelvanambi, M.; Dardani, I. P.; Emert, B.; Bodkin, N.; Braun, J.; Fingerman, D.; Kaur, A.; Jain, N.; Ravindran, P. T.; Mellis, I. A.; Kiani, K.; Alicea, G. M.; Fane, M. E.; Ahmed, S. S.; Li, H.; Chen, Y.; Chai, C.; Kaster, J.; Witt, R. G.; Lazcano, R.; Ingram, D. R.; Johnson, S. B.; Wani, K.; Dunagin, M. C.; Lazar, A. J.; Weeraratna, A. T.; Wargo, J. A.; Herlyn, M.; Raj, A. Diverse Clonal Fates Emerge upon Drug Treatment of Homogeneous Cancer Cells. *Nature* **2023**, *620* (7974), 651–659. https://doi.org/10.1038/s41586-023-06342-8.

(20) Gutierrez, C.; Al'Khafaji, A. M.; Brenner, E.; Johnson, K. E.; Gohil, S. H.; Lin, Z.; Knisbacher, B. A.; Durrett, R. E.; Li, S.; Parvin, S.; Biran, A.; Zhang, W.; Rassenti, L.; Kipps, T. J.; Livak, K. J.; Neuberg, D.; Letai, A.; Getz, G.; Wu, C. J.; Brock, A. Multifunctional Barcoding with ClonMapper Enables High-Resolution Study of Clonal Dynamics during Tumor Evolution and Treatment. *Nat. Cancer* **2021**, *2* (7), 758–772. https://doi.org/10.1038/s43018-021-00222-8.

(21) Holze, H.; Talarmain, L.; Fennell, K. A.; Lam, E. Y.; Dawson, M. A.; Vassiliadis, D. Analysis of Synthetic Cellular Barcodes in the Genome and Transcriptome with BARtab and Bartools. *Cell Rep. Methods* **2024**, 100763. https://doi.org/10.1016/j.crmeth.2024.100763.

(22) Sun, W.; Perkins, M.; Huyghe, M.; Faraldo, M. M.; Fre, S.; Perié, L.; Lyne, A.-M. Extracting, Filtering and Simulating Cellular Barcodes Using CellBarcode Tools. *Nat. Comput. Sci.* **2024**, *4* (2), 128–143. https://doi.org/10.1038/s43588-024-00595-7.

(23) *TeamPerie/CellBarcodeSim at v1*. GitHub. https://github.com/TeamPerie/CellBarcodeSim (accessed 2025-09-10).

(24) *barcodetrackR*. Bioconductor. http://bioconductor.org/packages/barcodetrackR/ (accessed 2025-09-10).

(25) Espinoza, D. A.; Mortlock, R. D.; Koelle, S. J.; Wu, C.; Dunbar, C. E. Interrogation of Clonal Tracking Data Using barcodetrackR. *Nat. Comput. Sci.* **2021**, *1* (4), 280–289. https://doi.org/10.1038/s43588-021-00057-4.

(26) Hadj Abed, L.; Tak, T.; Cosgrove, J.; Perié, L. CellDestiny: A RShiny Application for the Visualization and Analysis of Single-Cell Lineage Tracing Data. *Front. Med.* **2022**, *9*, 919345. https://doi.org/10.3389/fmed.2022.919345.
(27) TeamPerie/CellDestiny, 2025. https://github.com/TeamPerie/CellDestiny (accessed 2025-09-10).
(28) Vassiliadis, D. DaneVass/Bartools, 2025. https://github.com/DaneVass/bartools (accessed 2025-09-10).
(29) Akimov, Y.; Bulanova, D.; Timonen, S.; Wennerberg, K.; Aittokallio, T. Improved Detection of Differentially Represented DNA Barcodes for High-throughput Clonal Phenomics. *Mol. Syst. Biol.* **2020**, *16* (3), e9195. https://doi.org/10.15252/msb.20199195.
(30) Love, M. I.; Huber, W.; Anders, S. Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2. *Genome Biol.* **2014**, *15* (12), 550. https://doi.org/10.1186/s13059-014-0550-8.
(31) Robinson, M. D.; McCarthy, D. J.; Smyth, G. K. edgeR: A Bioconductor Package for Differential Expression Analysis of Digital Gene Expression Data. *Bioinformatics* **2010**, *26* (1), 139–140. https://doi.org/10.1093/bioinformatics/btp616.
(32) *barbieQ*. Bioconductor. http://bioconductor.org/packages/barbieQ/ (accessed 2025-09-10).
(33) Wu, C.; Espinoza, D. A.; Koelle, S. J.; Yang, D.; Truitt, L.; Schlums, H.; Lafont, B. A.; Davidson-Moncada, J. K.; Lu, R.; Kaur, A.; Hammer, Q.; Li, B.; Panch, S.; Allan, D. A.; Donahue, R. E.; Childs, R. W.; Romagnani, C.; Bryceson, Y. T.; Dunbar, C. E. Clonal Expansion and Compartmentalized Maintenance of Rhesus Macaque NK Cell Subsets. *Sci. Immunol.* **2018**, *3* (29), eaat9781. https://doi.org/10.1126/sciimmunol.aat9781.
(34) Ritchie, M. E.; Phipson, B.; Wu, D.; Hu, Y.; Law, C. W.; Shi, W.; Smyth, G. K. Limma Powers Differential Expression Analyses for RNA-Sequencing and Microarray Studies. *Nucleic Acids Res.* **2015**, *43* (7), e47. https://doi.org/10.1093/nar/gkv007.
(35) Law, C. W.; Alhamdoosh, M.; Su, S.; Dong, X.; Tian, L.; Smyth, G. K.; Ritchie, M. E. RNA-Seq Analysis Is Easy as 1-2-3 with Limma, Glimma and edgeR. F1000Research December 28, 2018. https://doi.org/10.12688/f1000research.9005.3.
(36) Benjamini, Y.; Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Ser. B Methodol.* **1995**, *57* (1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.