nature portfolio

https://doi.org/10.1038/s41587-021-00870-2

Supplementary information

Modular, efficient and constant-memory single-cell RNA-seq preprocessing

In the format provided by the
authors and unedited

Supplementary Figures

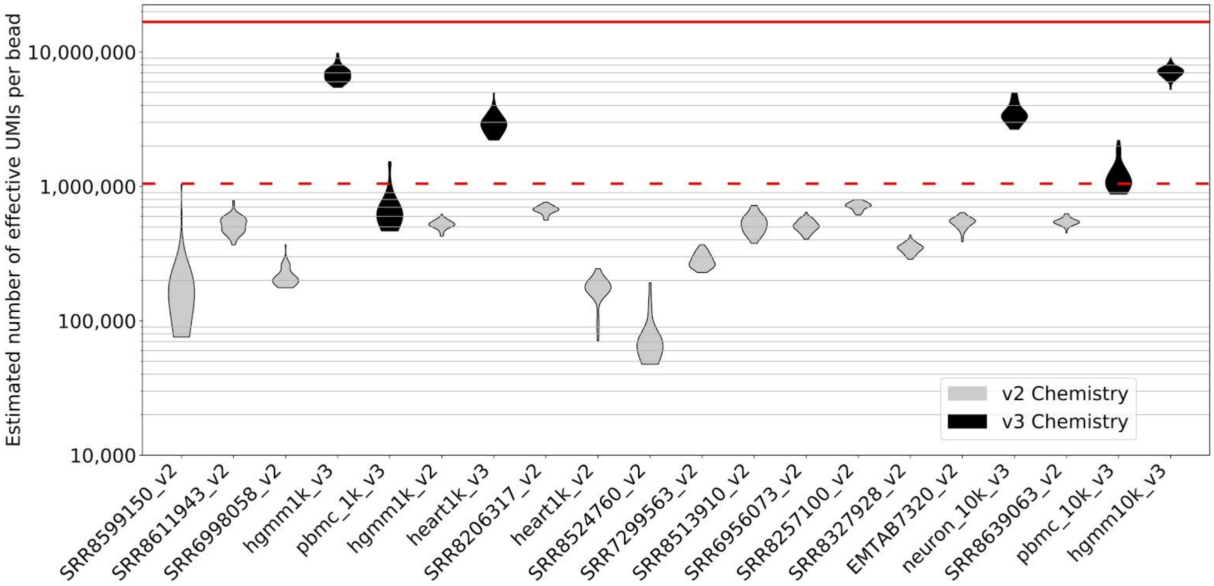
Modular, efficient, and constant-memory single-cell RNA-seq pre-processing

Páll Melsted[1,*], A. Sina Booeshaghi[2,*], Lauren Liu[3], Fan Gao[4,5], Lambda Lu[4], Kyung Hoi (Joseph) Min[6], Eduardo da Veiga Beltrame[4], Kristján Eldjárn Hjörleifsson[3] , Jase Gehring[7], and Lior Pachter[3,4]

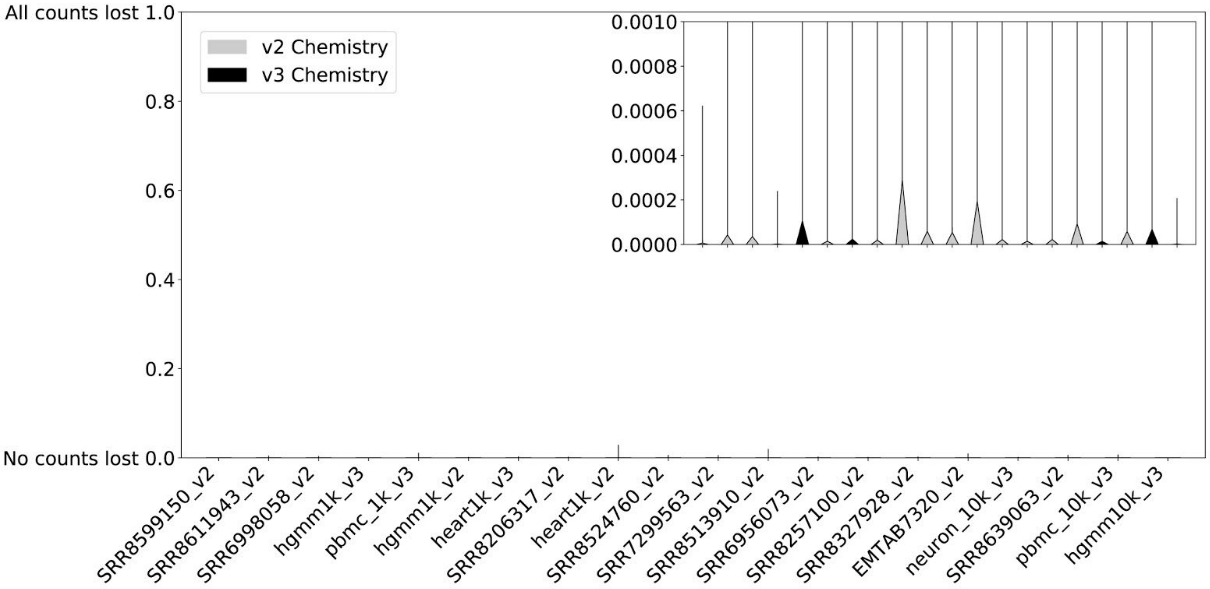
1. Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, Reykjavik, Iceland
2. Department of Mechanical Engineering, California Institute of Technology, Pasadena, California
3. Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California
4. Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, California
5. Bioinformatics Resource Center, Beckman Institute, California Institute of Technology, Pasadena, California
6. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts
7. Department of Genome Science, University of Washington, Seattle, Washington

Address correspondence to lpachter@caltech.edu

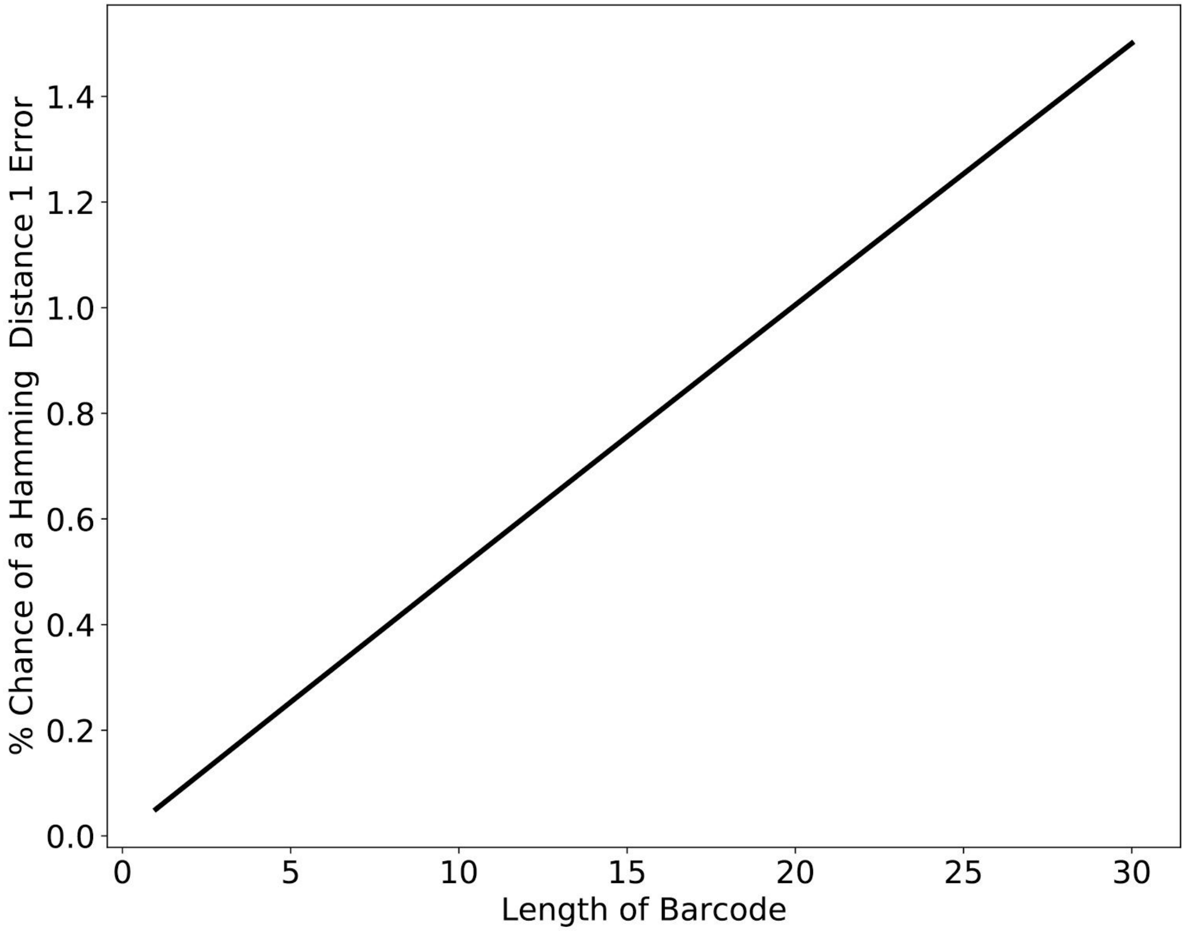
* Authors contributed equally



Supplementary Figure 1.1: Estimates of the effective number of UMIs per bead for each of the benchmark panel datasets, determined from observed collisions of UMIs across unique genes and assuming UMIs are sampled uniformly with replacement (see Supplementary Note for further details). The dashed red line is the theoretical maximum for the number of UMIs on a v2 chemistry bead (4^{10}=1,048,576) and the red line is the theoretical maximum for the number of UMIs on a v3 chemistry bead (4^{12}=16,777,216). The datasets are ordered by number of reads. UMI pools from 10x Chromium v2 and v3 chemistry are found to be highly complex, with the effective number of UMIs approaching the theoretical maximum in many cases. Our estimates for UMI complexity vary across experiments; this could be due to batch effects, or model misspecification. Sequencing chimeras could also affect UMI complexity estimates, specifically estimates would be increased with more chimeras. This would reduce the estimates of intra-gene collisions due to naïve collapsing.

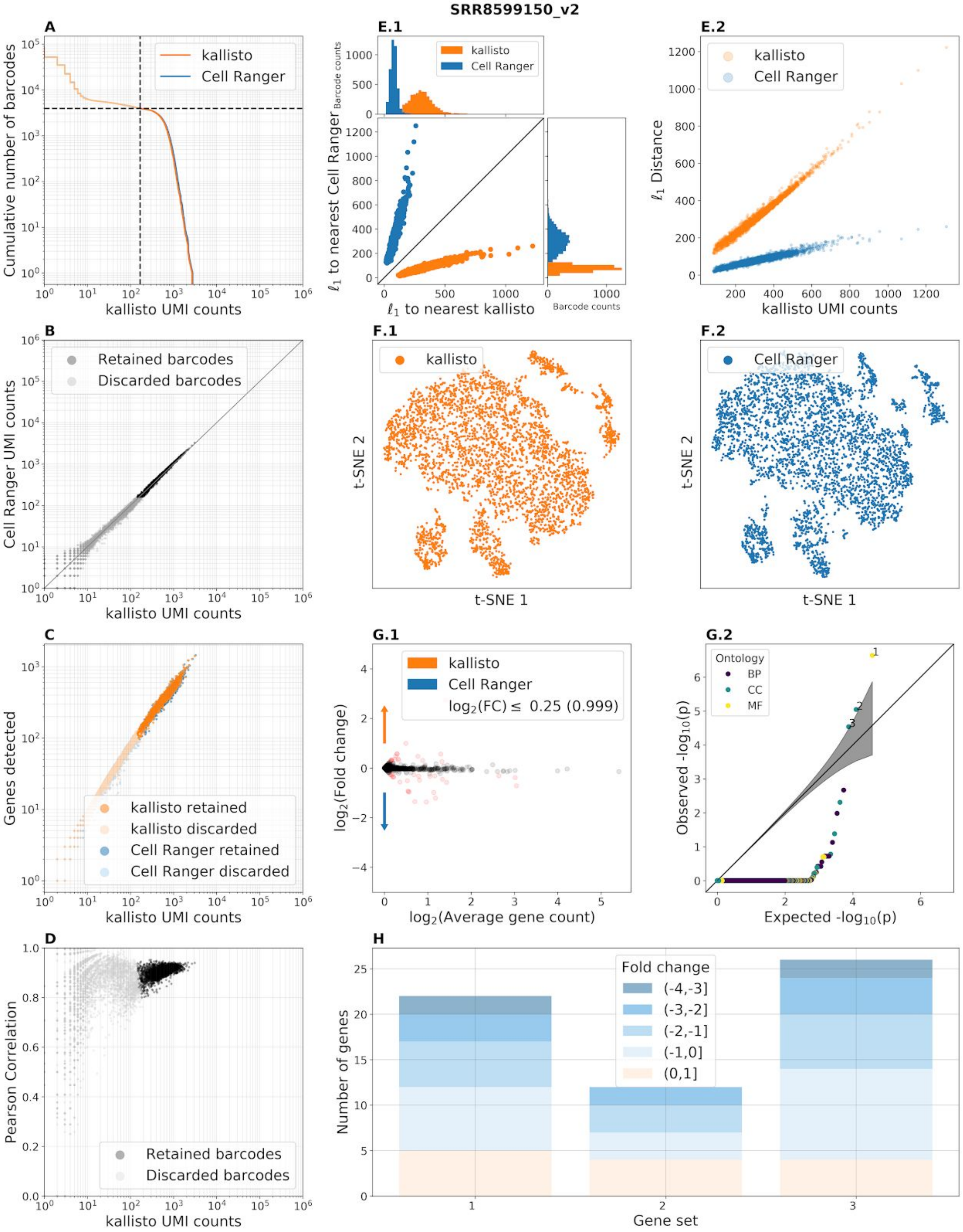


Supplementary Figure 1.2: Fraction of UMIs lost per gene across cells in the benchmark panel due to over-collapsing. The inset figure is a magnified view of the violin plot demonstrating that the fraction of counts lost due to UMI collapsing at the gene level is on the order of 0.0002.

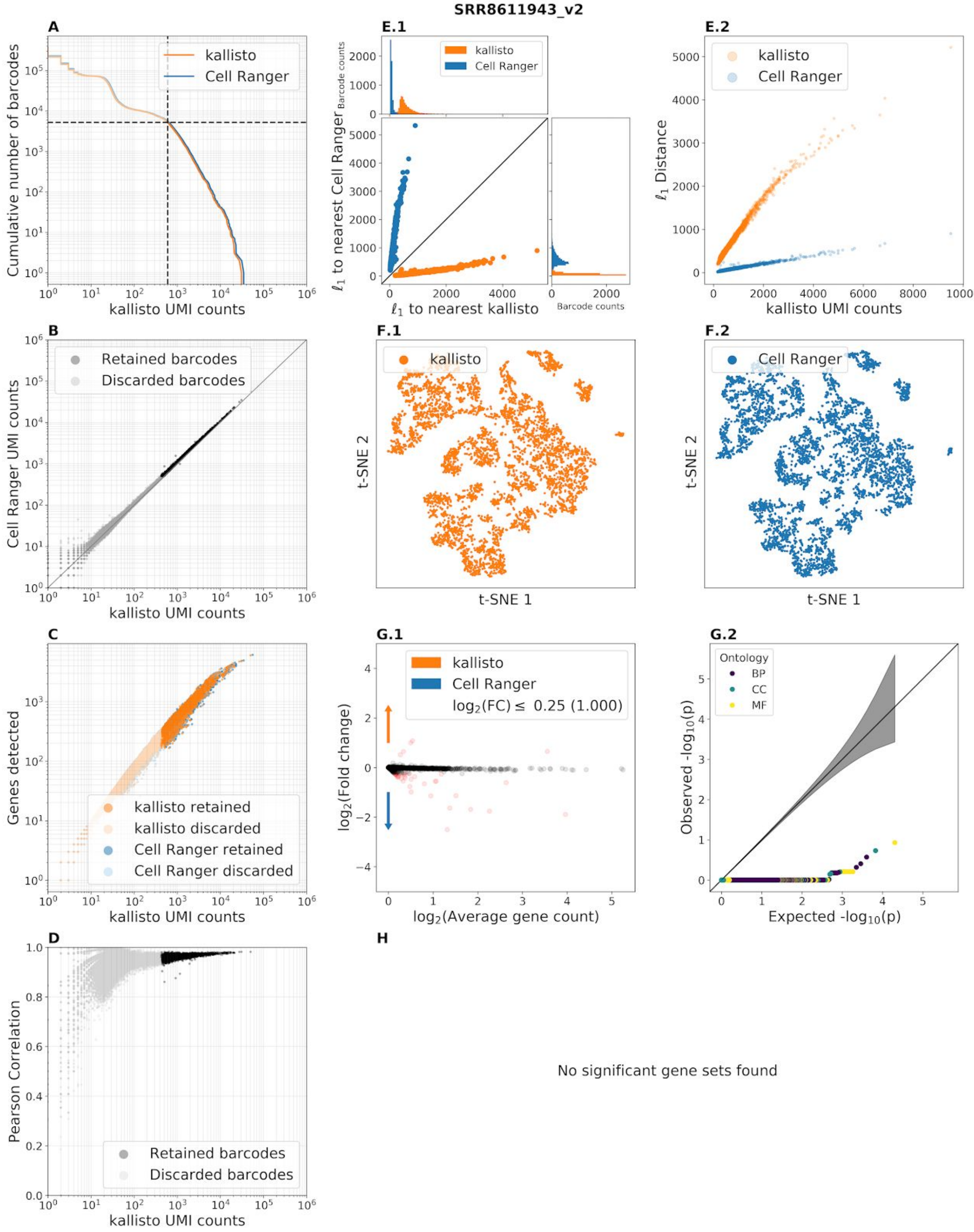


Supplementary Figure 2: The expected percentage of Barcodes (or UMIs) that will have one error and can therefore be corrected with a Hamming distance 1 correction algorithm. The y-axis displays the value of the function $f(L)$ where $f(L) = 100 \cdot L\hat{p}(1-\hat{p})^{L-1}$, and where $\hat{p}$ is the per base sequencing error probability estimated by averaging the error estimates across all the datasets in the benchmark panel (Supplementary Table 2).

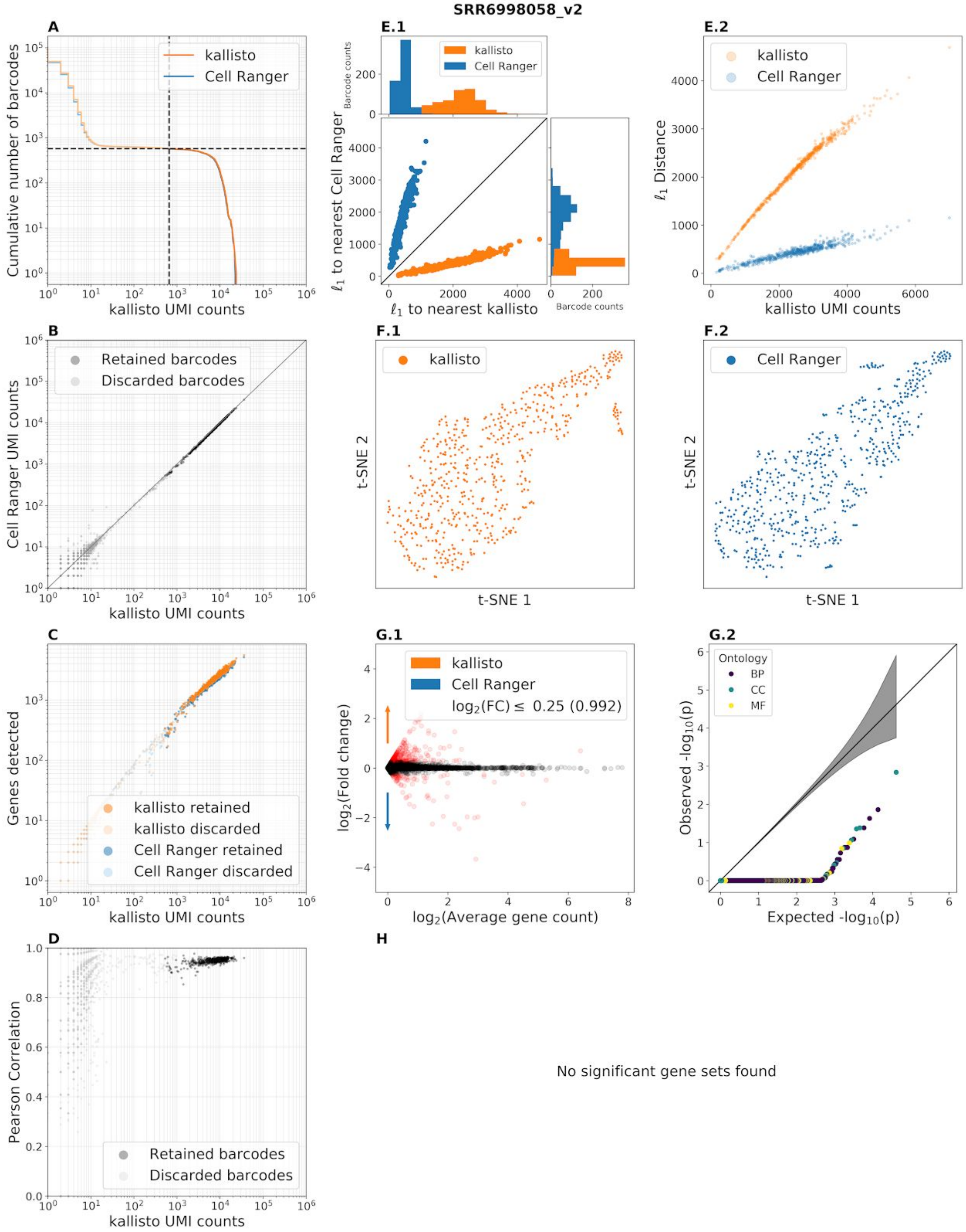
All **Supplementary Figure 3** benchmark panels are configured as follows: (A) “Knee plots” for kallisto and Cell Ranger showing, for a given UMI count (x-axis), the number of cells that contain at least that many UMI counts (y-axis). The dashed lines correspond to the Cell Ranger filtered cells. (B) Correspondence in the number of distinct UMIs per cell between the workflows. (C) Genes detected by kallisto and Cell Ranger as a function of distinct UMI counts per cell. (D) Pearson correlation between gene counts as a function of the distinct UMI counts per cell. (E.1) The l_1 distance between gene abundances for each kallisto cell and its nearest neighbor plotted against each kallisto cell and its corresponding Cell Ranger cell (orange) and the l_1 distance between the gene abundances for each Cell Ranger cell and its nearest neighbor plotted against each Cell Ranger cell and its corresponding kallisto cell (orange). Marginal distributions show that each kallisto cell is closest to its corresponding Cell Ranger cell and that each Cell Ranger cell is closest to its corresponding kallisto cell. (E.2) l_1 distance between kallisto and Cell Ranger cells as a function of UMI counts. (F.1) kallisto t-SNE from the first 10 principal components. (F.2) Cell Ranger t-SNE from the first 10 principal components. (G.1) MA plot for all genes between kallisto and Cell Ranger. Most of the genes have a $log_2(FC) \leq 0.25$ (G.2) QQ plot comparing the distribution of observed distribution of p-values of GSEA, after Bonferroni correction for multiple testing across ontologies and datasets, with the expected distribution of a uniform distribution between 0 and 1. If the observed distribution does not significantly deviate from the expected distribution, then the points should lie close to the diagonal line, $y = x$. The gray ribbon around the line is the 95% confidence interval. Here most GO terms have adjusted $p = 1$, meaning that most GO terms are very depleted of genes “differentially expressed (DE)” between the kallisto and Cell Ranger matrices. GO terms above $y = x$ are labeled. Generally, GO terms significantly enriched among “DE” genes are related to ribosomal proteins and are labeled by numbers corresponding to GO terms in the figure caption. The points are also colored by ontology: biological processes (BP), cellular components (CC), and molecular functions (MF). (H) Significant differential gene sets between Cell Ranger and kallisto.


SRR8599150_v2
A
Cumulative number of barcodes
kallisto
Cell Ranger
kallisto UMI counts
E.1
Barcode counts
ℓ1 to nearest Cell Ranger
ℓ1 to nearest kallisto
Barcode counts
E.2
ℓ1 Distance
kallisto UMI counts
B
Retained barcodes
Discarded barcodes
Cell Ranger UMI counts
kallisto UMI counts
F.1
kallisto
t-SNE 2
t-SNE 1
F.2
Cell Ranger
t-SNE 2
t-SNE 1
C
Genes detected
kallisto retained
kallisto discarded
Cell Ranger retained
Cell Ranger discarded
kallisto UMI counts
G.1
kallisto
Cell Ranger
log2(FC) ≤ 0.25 (0.999)
log2(Fold change)
log2(Average gene count)
G.2
Ontology
BP
CC
MF
Observed -log10(p)
Expected -log10(p)
1
2
3
D
Pearson Correlation
Retained barcodes
Discarded barcodes
kallisto UMI counts
H
Fold change
(-4,-3]
(-3,-2]
(-2,-1]
(-1,0]
(0,1]
Number of genes
Gene set
1
2
3

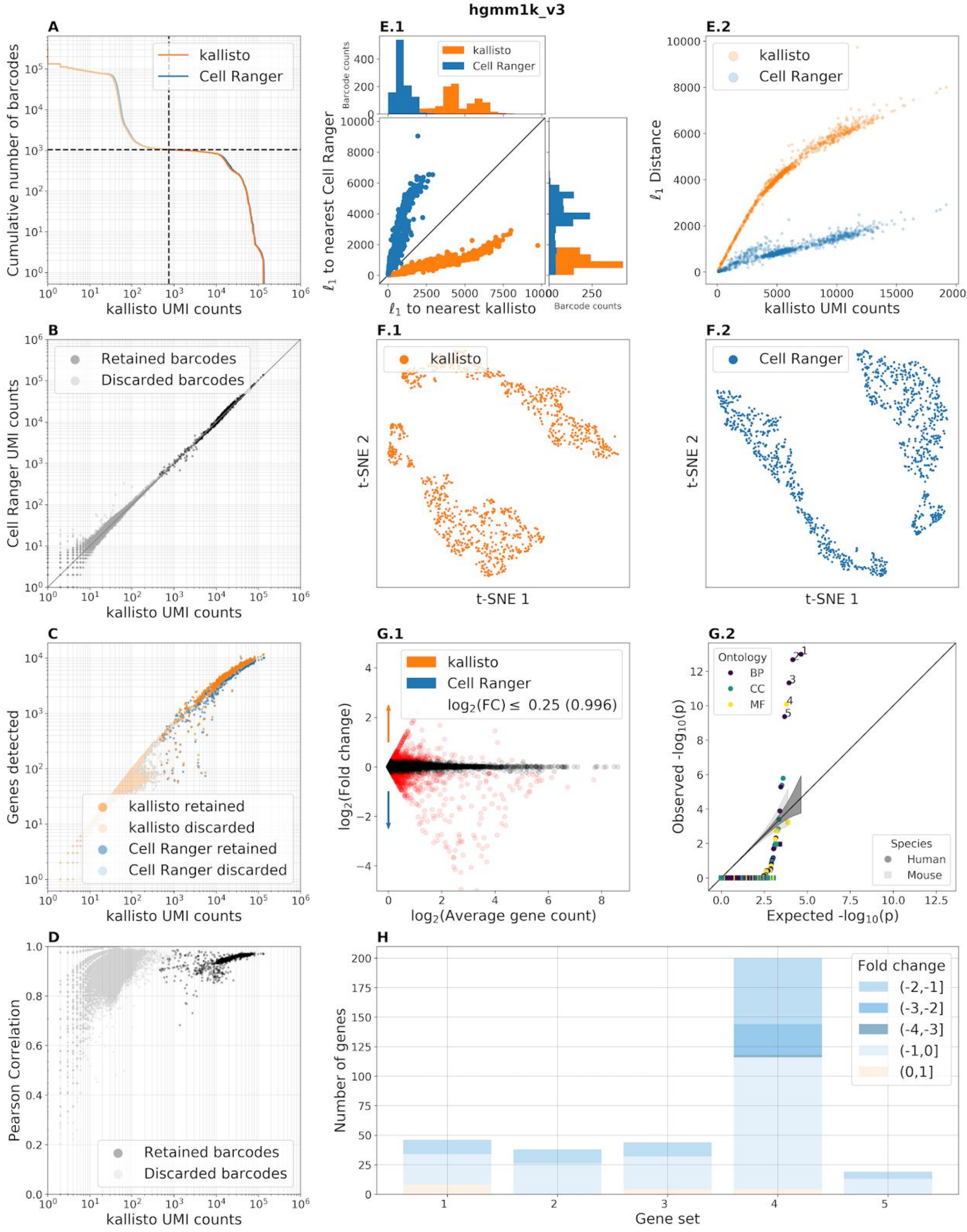
Supplementary Figure 3.1: Benchmark panel of data from O’Koren et al. 2019 (O’Koren et al., 2019) (SRR8599150). Enriched GO terms are 1-structural constituent of ribosome, 2-cytosolic small ribosomal subunit, 3-cytosolic large ribosomal subunit.



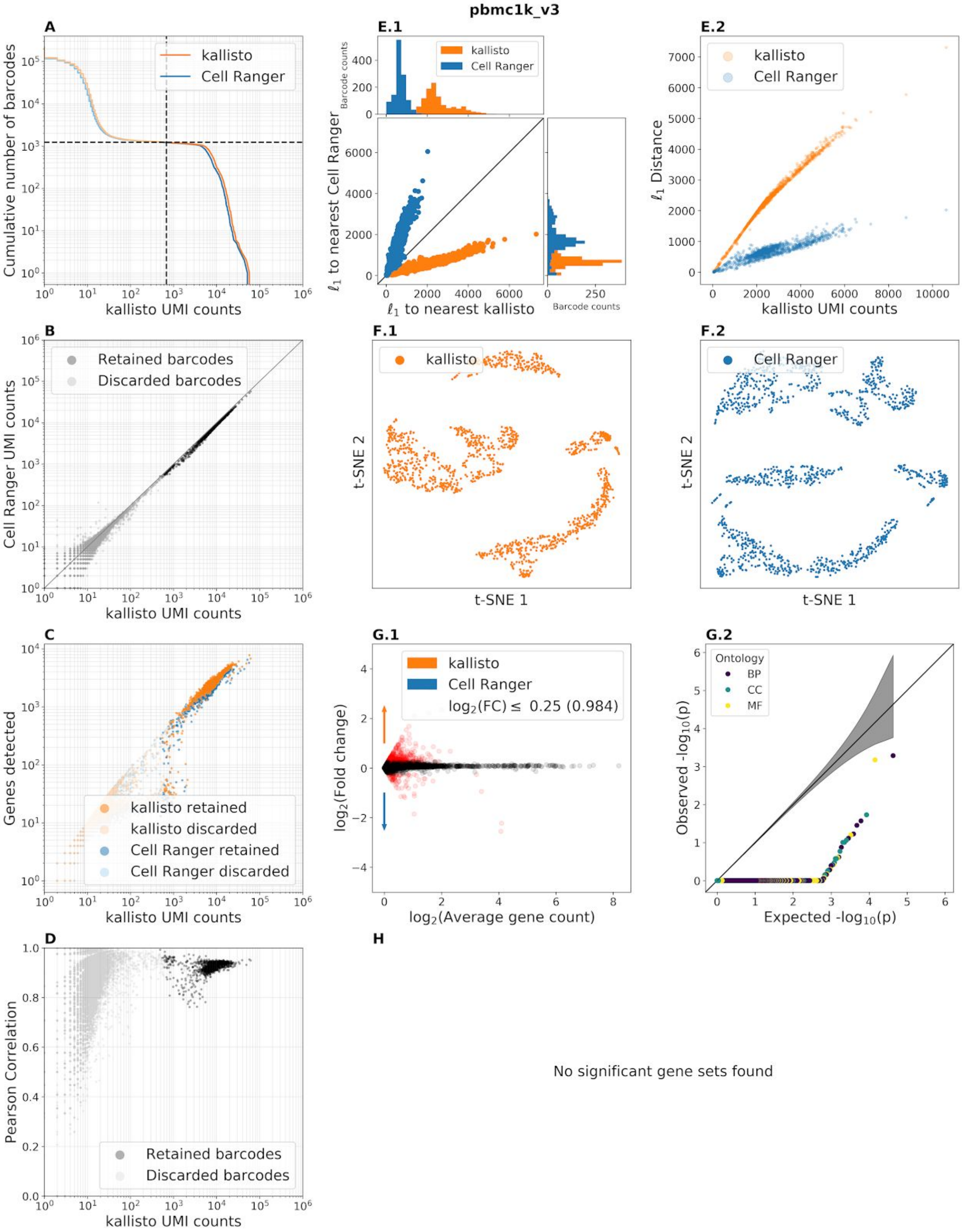
Supplementary Figure 3.2: Benchmark panel of data from Packer et al. 2019 (Packer et al., 2019) (SRR8611943).



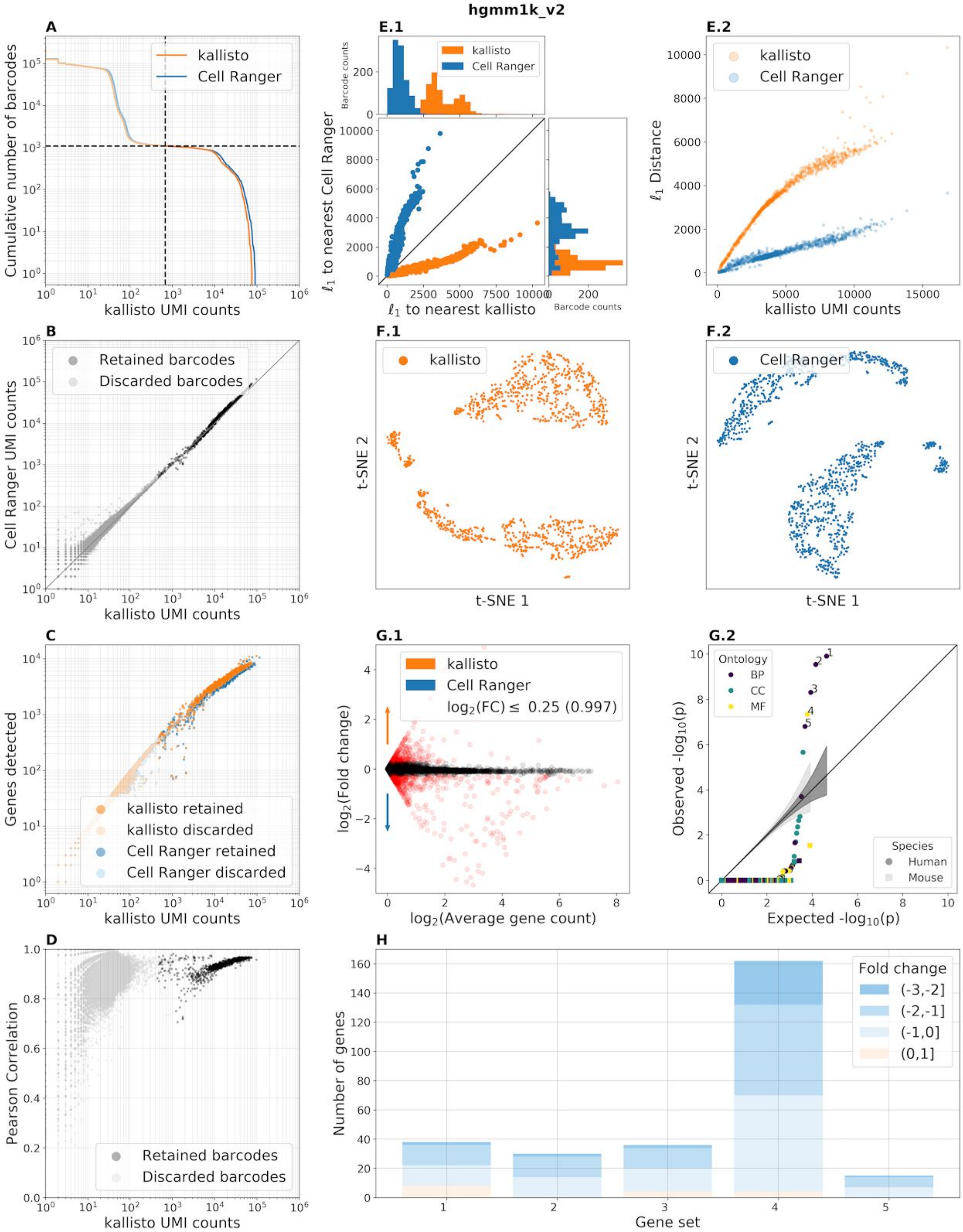
Supplementary Figure 3.3: Benchmark panel of data from Jin et al. 2018 (Jin et al., 2018) (SRR6998058).



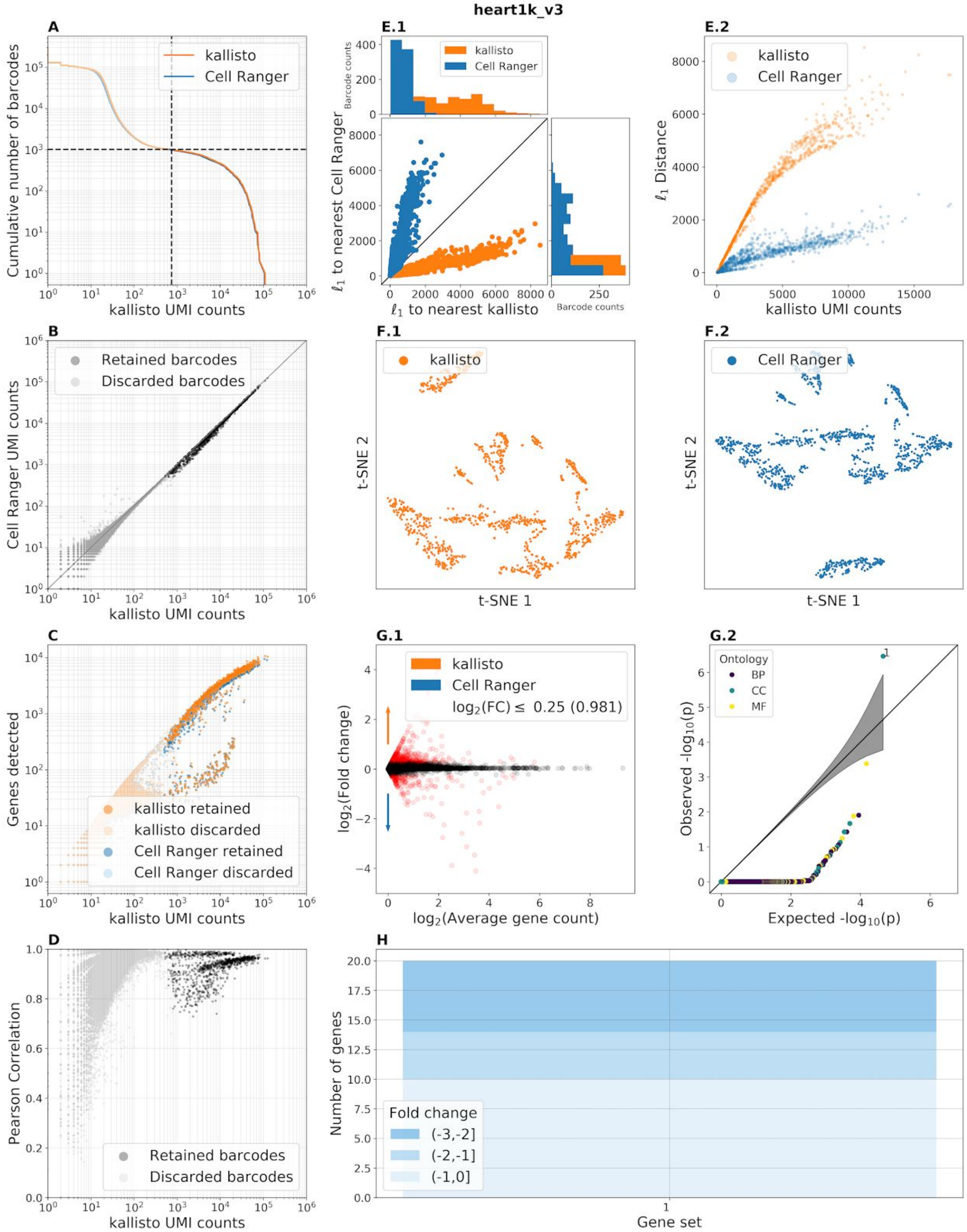
Supplementary Figure 3.4: Benchmark panel of the 10x Genomics hgmm1k_v3 dataset. Enriched GO terms are 1-nuclear-transcribed mRNA catabolic process, nonsense-mediated decay, 2-SRP-dependent cotranslational protein targeting to membrane, 3-translational initiation, 4-structural constituent of ribosome, 5-viral transcription.



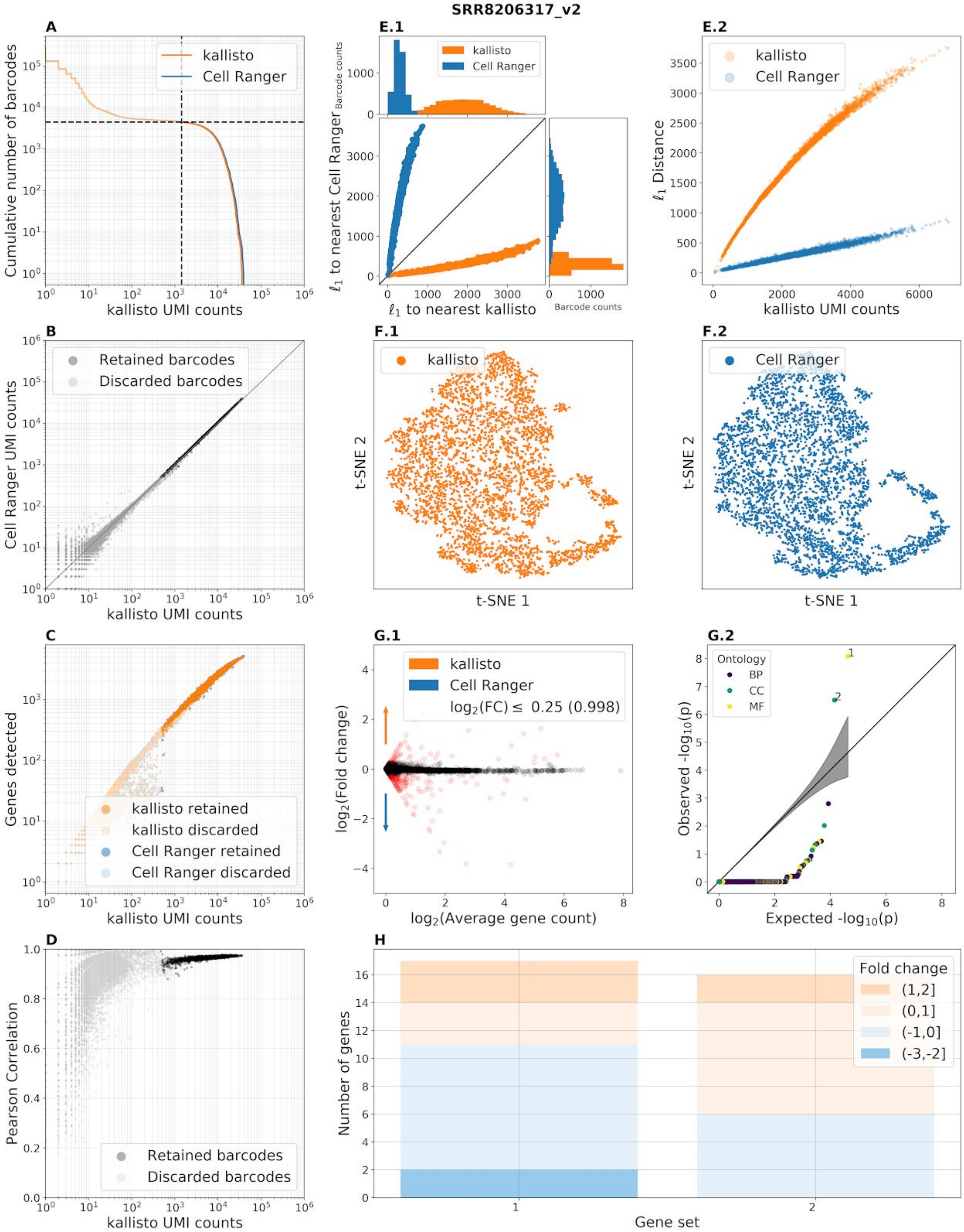
Supplementary Figure 3.5: Benchmark panel of the 10x Genomics pbmc1k_v3 dataset.



Supplementary Figure 3.6: Benchmark panel of the 10x Genomics hgmm1k_v2 dataset. Enriched GO terms are 1-nuclear-transcribed mRNA catabolic process, nonsense-mediated decay, 2-SRP-dependent cotranslational protein targeting to membrane, 3-translational initiation, 4-structural constituent of ribosome, 5-viral transcription.

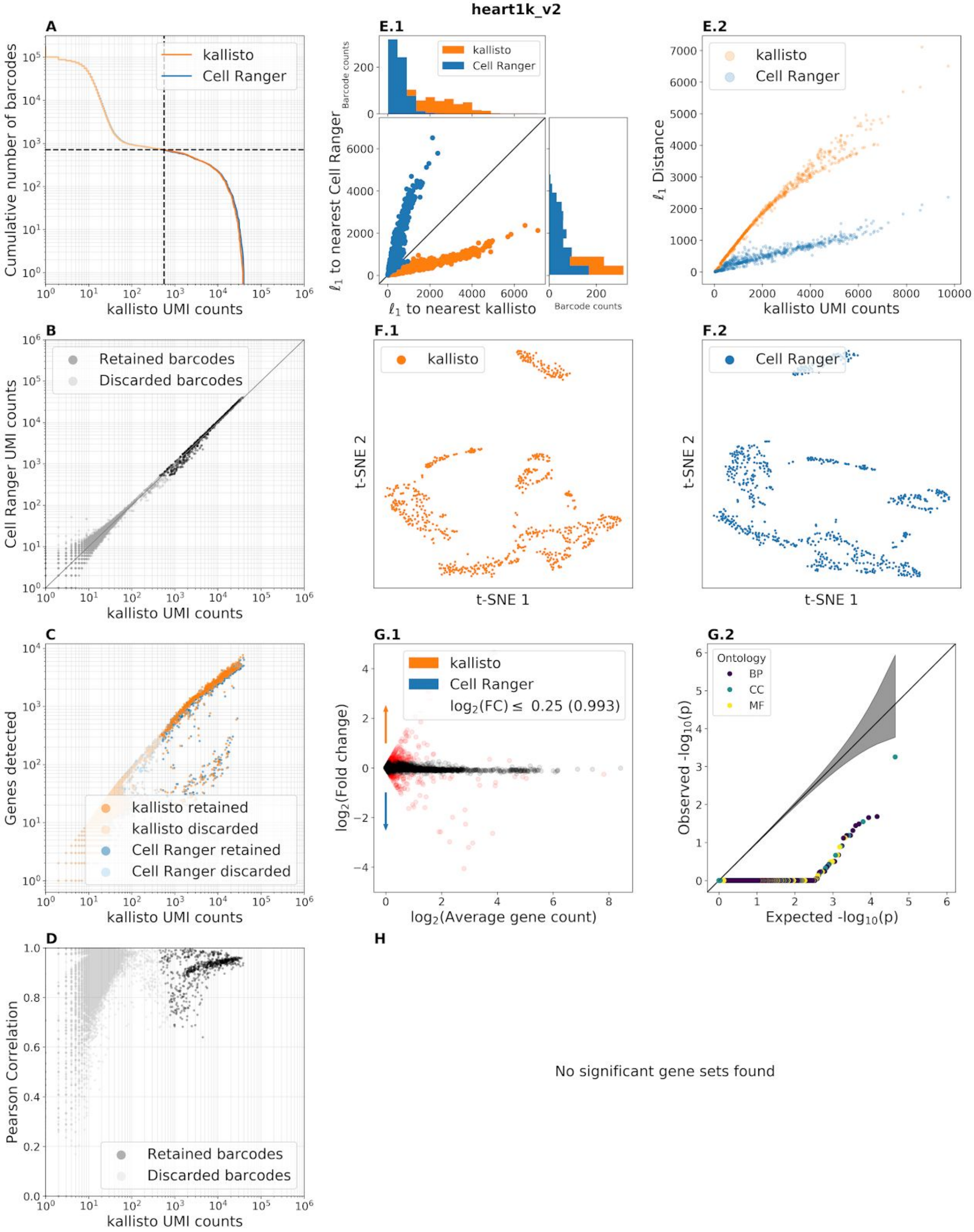


Supplementary Figure 3.7: Benchmark panel of the 10x Genomics heart1k_v3 dataset. Enriched GO terms are 1-cytosolic large ribosomal subunit.

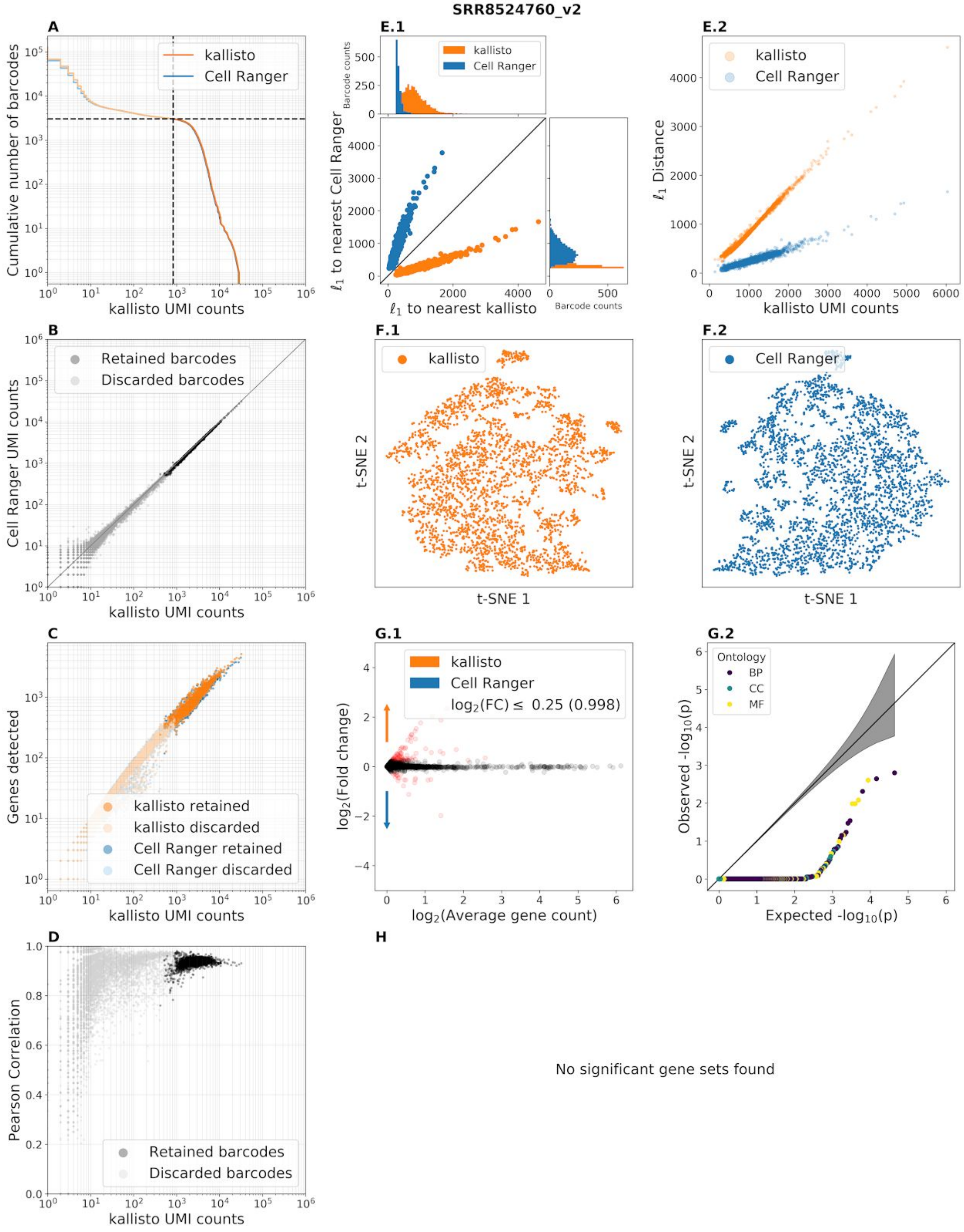


Supplementary Figure 3.8: Benchmark panel of data from Miller et al. 2019 (Miller et al., 2019)

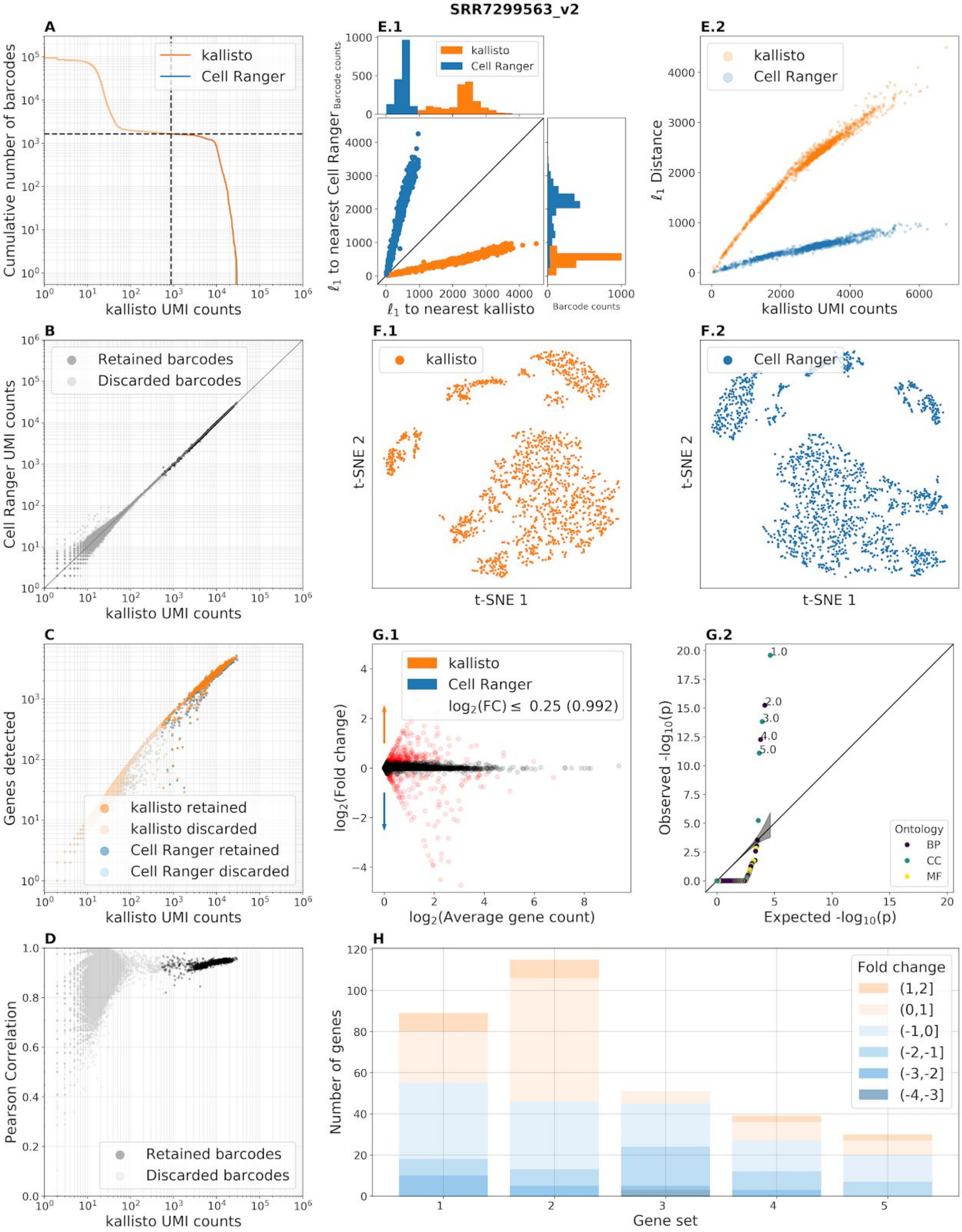
(SRR8206317). Enriched GO terms are 1-structural constituent of ribosome, 2-cytosolic large ribosomal subunit.



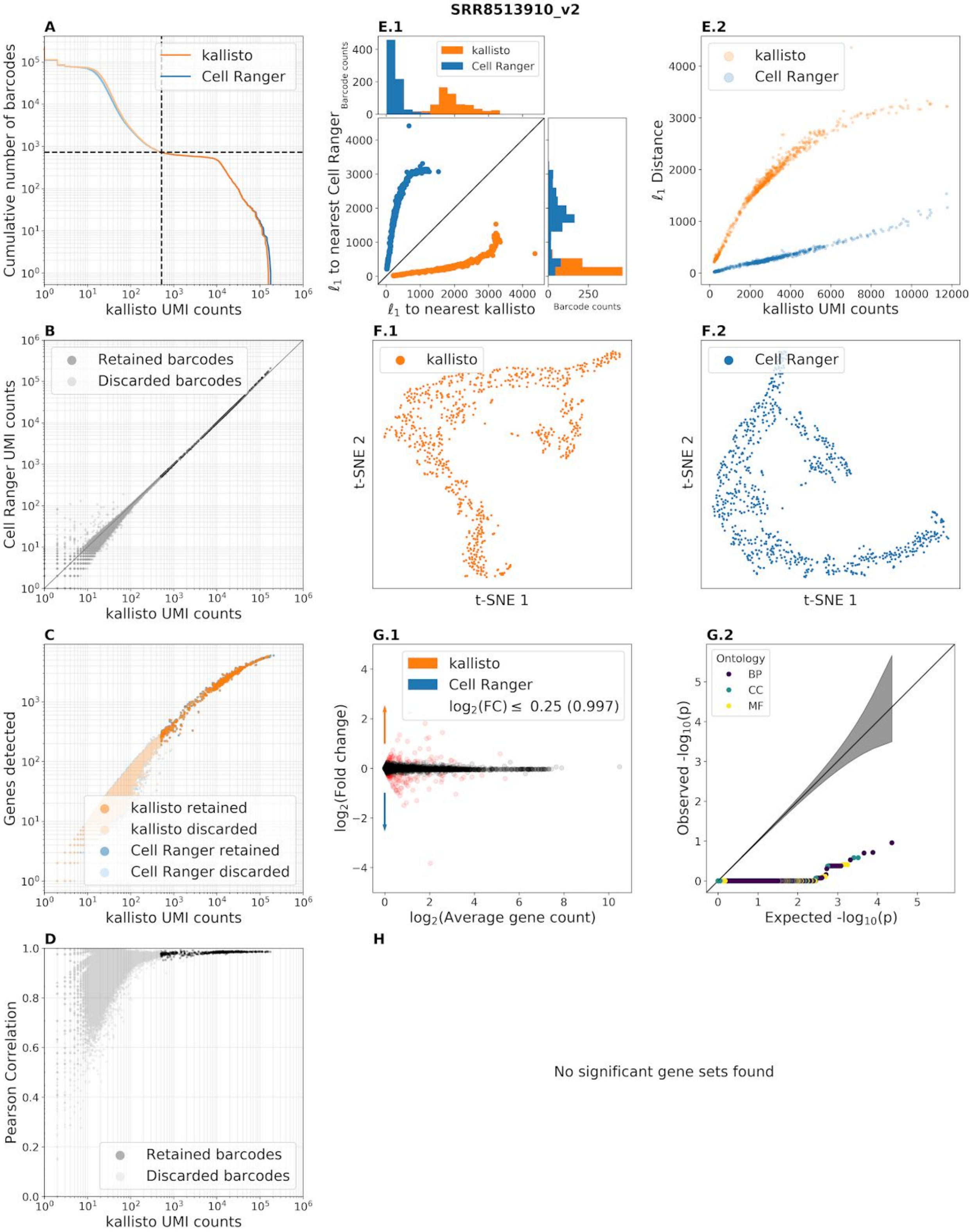
Supplementary Figure 3.9: Benchmark panel of the 10x Genomics heart1k_v2 dataset.



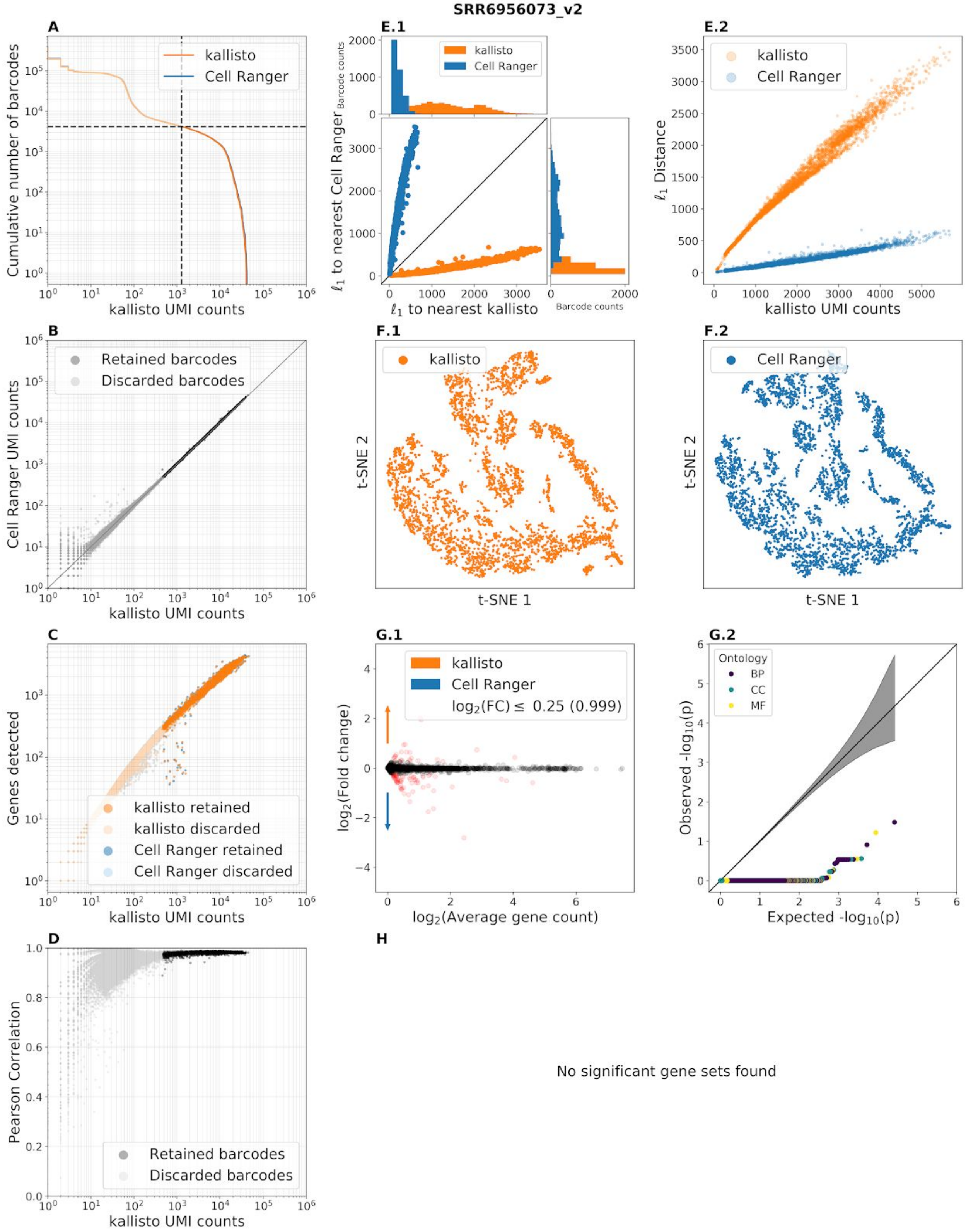
Supplementary Figure 3.10: Benchmark panel of data from Carosso et al. 2019 (Carosso et al., 2018) (SRR8524760).


SRR7299563_v2
A
kallisto
Cell Ranger
Cumulative number of barcodes
kallisto UMI counts
E.1
kallisto
Cell Ranger
Barcode counts
ℓ_1 to nearest Cell Ranger
ℓ_1 to nearest kallisto
Barcode counts
E.2
kallisto
Cell Ranger
ℓ_1 Distance
kallisto UMI counts
B
Retained barcodes
Discarded barcodes
Cell Ranger UMI counts
kallisto UMI counts
F.1
kallisto
t-SNE 2
t-SNE 1
F.2
Cell Ranger
t-SNE 2
t-SNE 1
C
kallisto retained
kallisto discarded
Cell Ranger retained
Cell Ranger discarded
Genes detected
kallisto UMI counts
G.1
kallisto
Cell Ranger
$\log_2$(FC) ≤ 0.25 (0.992)
$\log_2$(Fold change)
$\log_2$(Average gene count)
G.2
1.0
2.0
3.0
4.0
5.0
Ontology
BP
CC
MF
Observed $-\log_{10}$(p)
Expected $-\log_{10}$(p)
D
Retained barcodes
Discarded barcodes
Pearson Correlation
kallisto UMI counts
H
Fold change
(1,2]
(0,1]
(-1,0]
(-2,-1]
(-3,-2]
(-4,-3]
Number of genes
Gene set

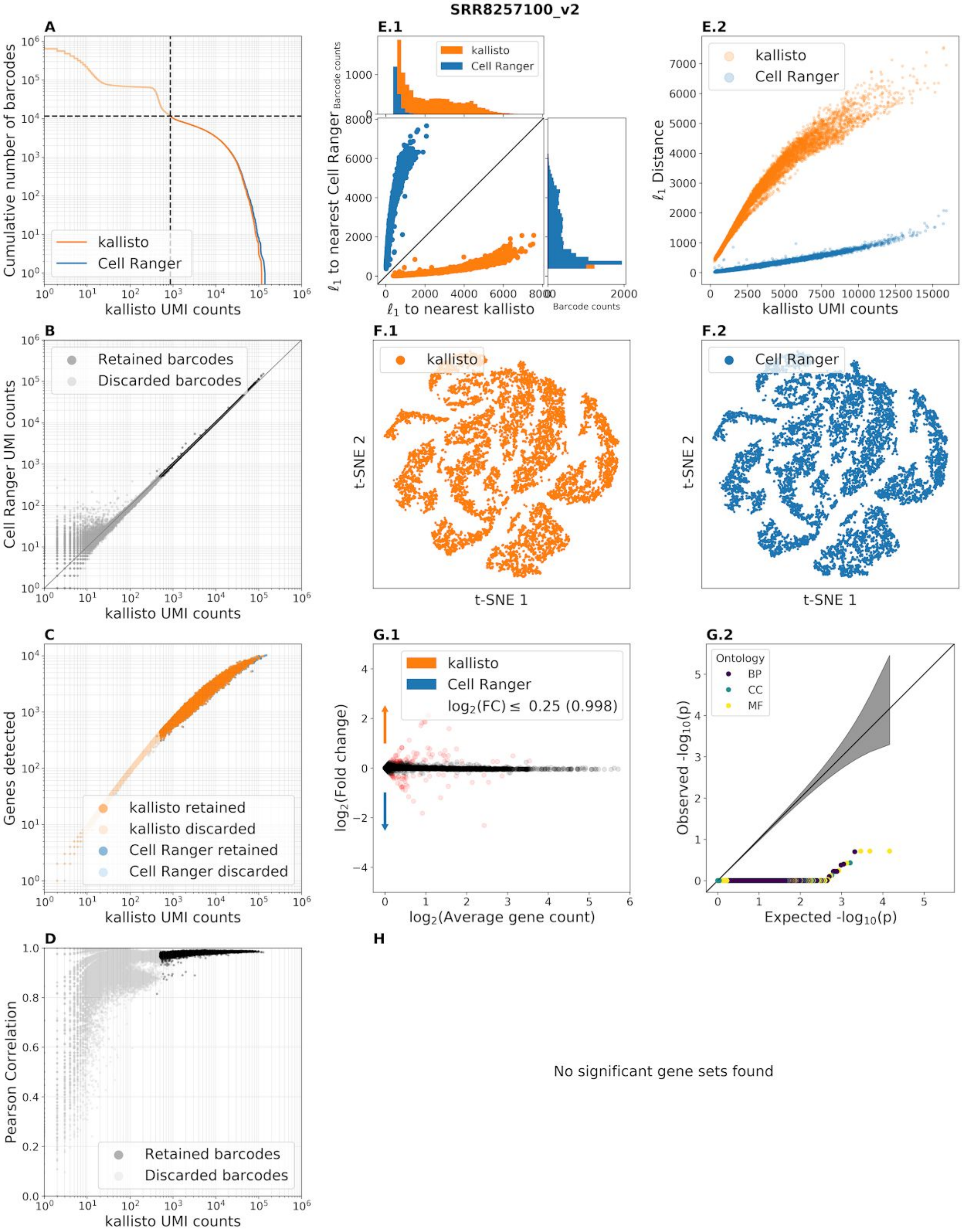
Supplementary Figure 3.11: Benchmark panel of data from Mays et al. 2018 (Mays et al., 2018) (SRR7299563). Enriched GO terms are 1-cytosolic large ribosomal subunit, 2-translation, 3-cytosolic small ribosomal subunit, 4-cytoplasmic translation, 5-polysomal ribosome.



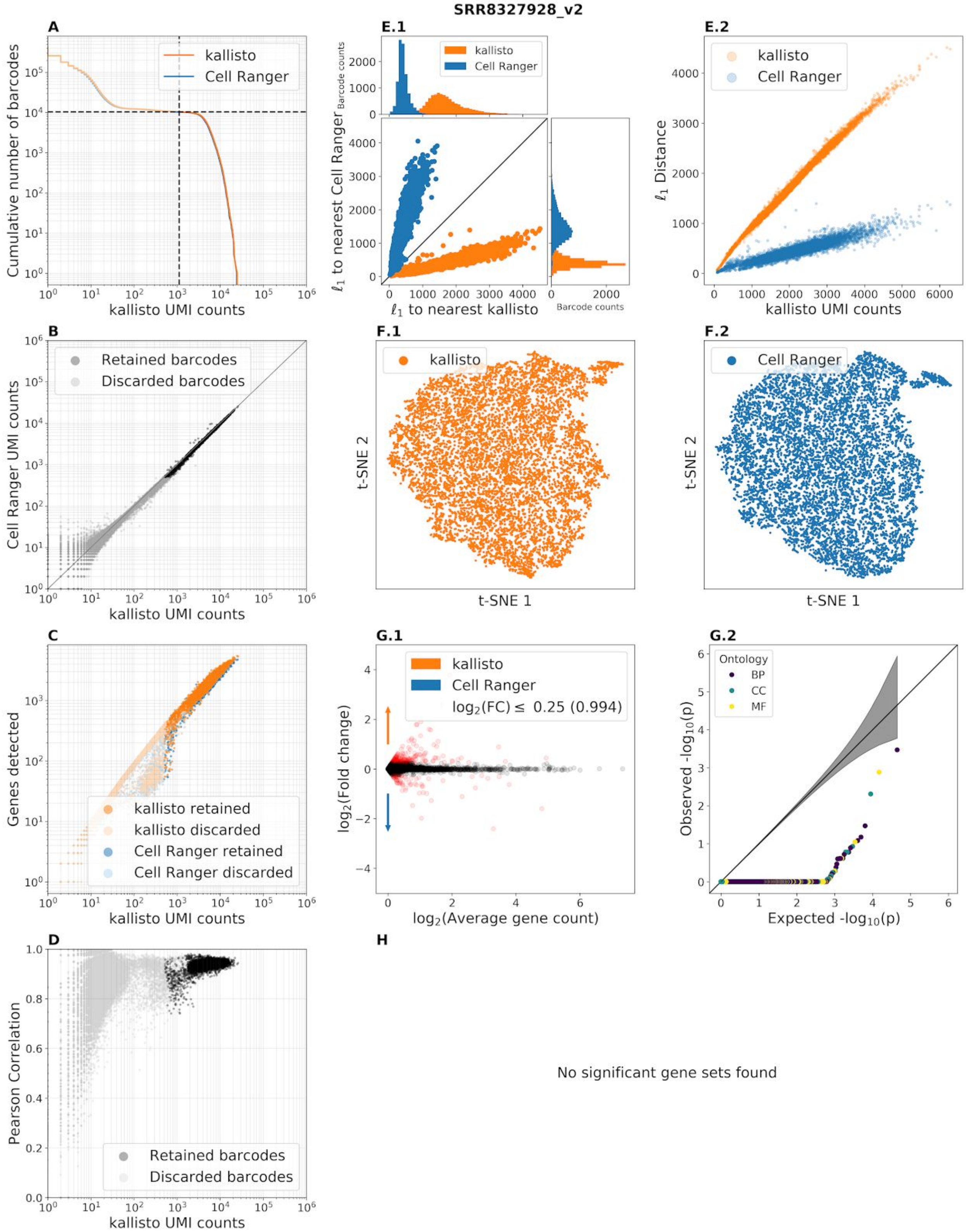
Supplementary Figure 3.12: Benchmark panel of data from the gene expression omnibus(Mahadevaraju et al., 2019) (SRR8513910).



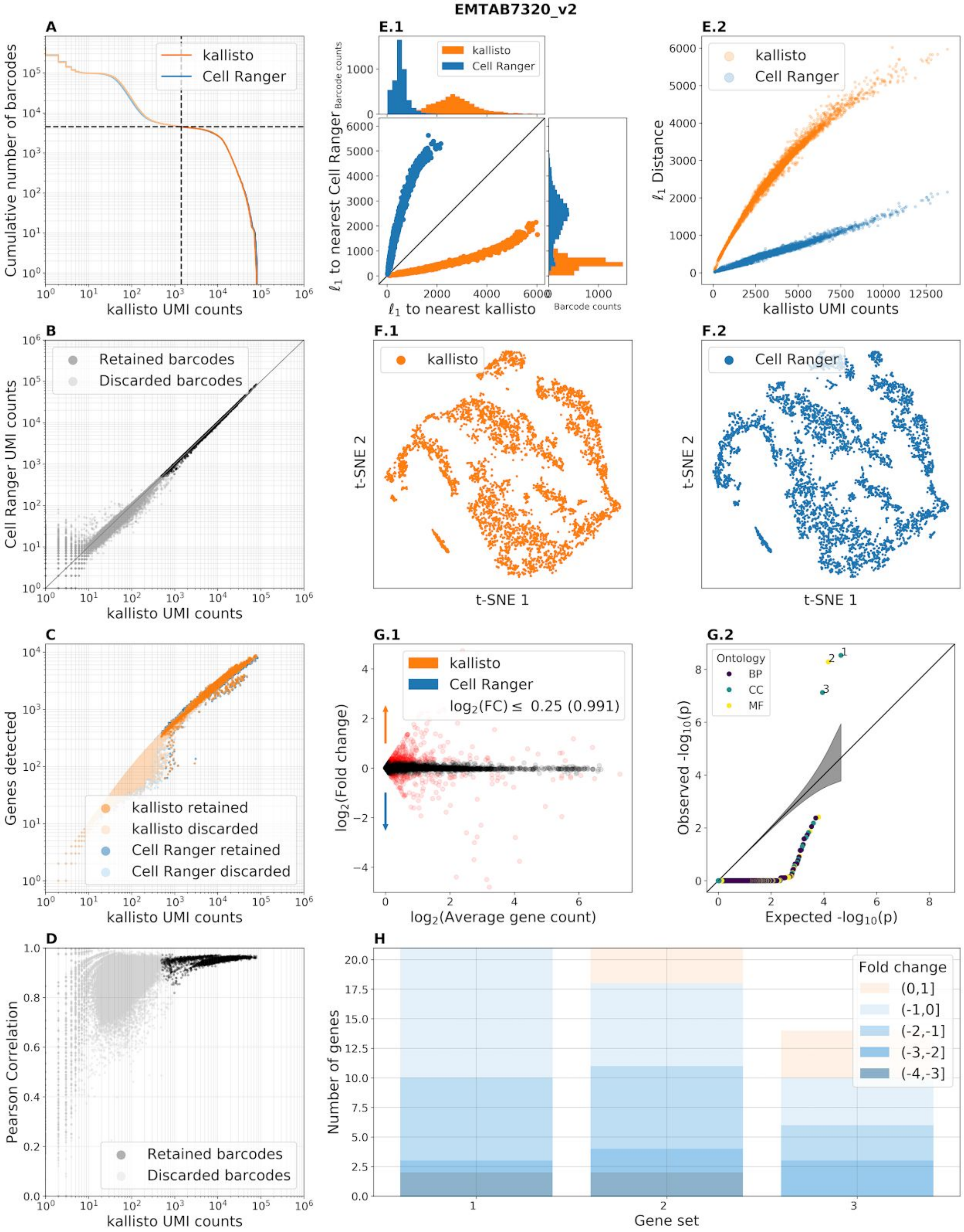
Supplementary Figure 3.13: Benchmark panel of data from Farrell et al. 2018(Farrell et al., 2018) (SRR6956073).



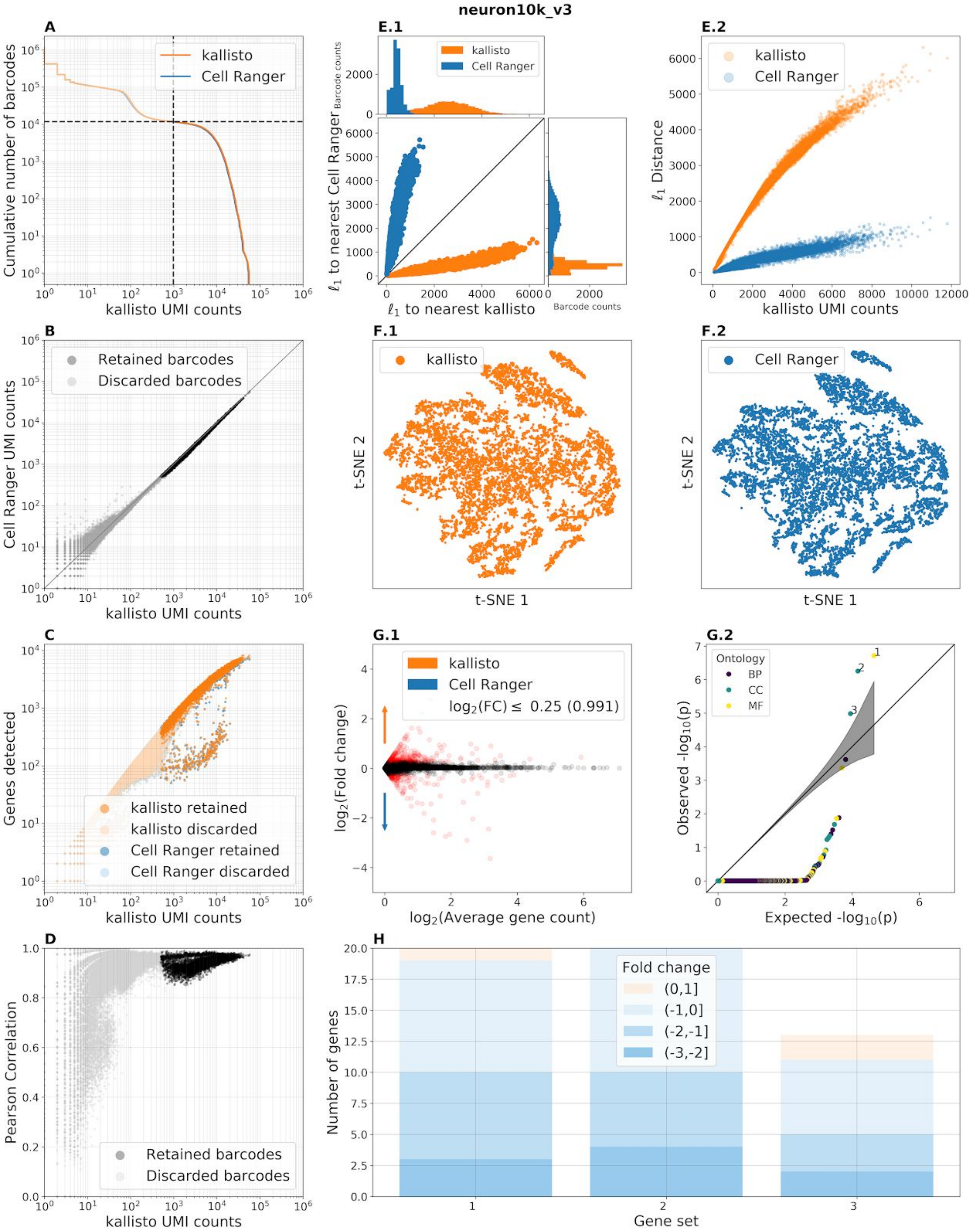
Supplementary Figure 3.14: Benchmark panel of data from Ryu et al. 2019(Ryu et al., 2019) (SRR8257100).



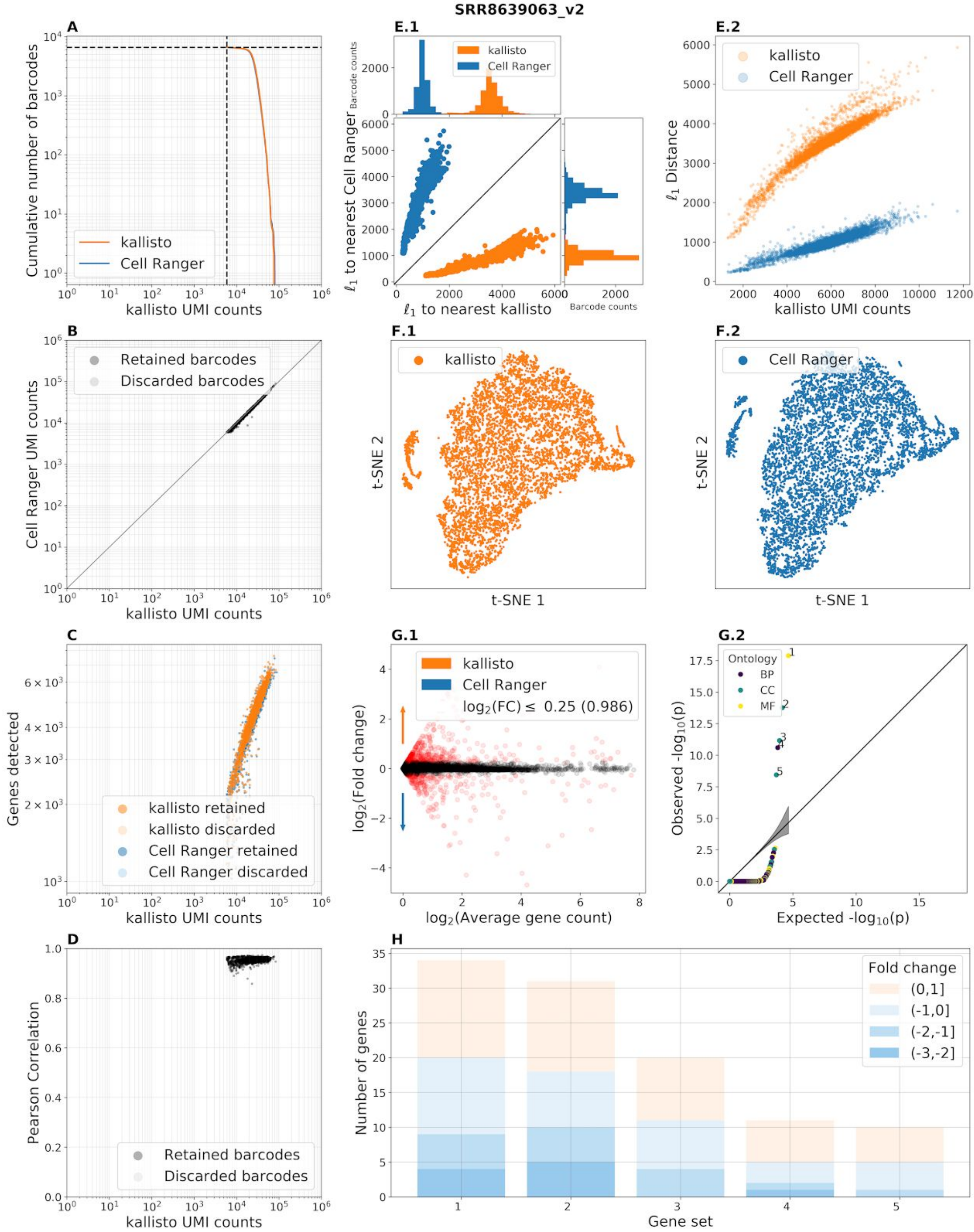
Supplementary Figure 3.15: Benchmark panel of data from Merino et al. 2019(Merino et al., 2019) (SRR8327928).


EMTAB7320_v2
A
kallisto
Cell Ranger
Cumulative number of barcodes
kallisto UMI counts
E.1
kallisto
Cell Ranger
Barcode counts
ℓ_1 to nearest Cell Ranger
ℓ_1 to nearest kallisto
Barcode counts
E.2
kallisto
Cell Ranger
ℓ_1 Distance
kallisto UMI counts
B
Retained barcodes
Discarded barcodes
Cell Ranger UMI counts
kallisto UMI counts
F.1
kallisto
t-SNE 2
t-SNE 1
F.2
Cell Ranger
t-SNE 2
t-SNE 1
C
kallisto retained
kallisto discarded
Cell Ranger retained
Cell Ranger discarded
Genes detected
kallisto UMI counts
G.1
kallisto
Cell Ranger
$\log_2$(FC) ≤ 0.25 (0.991)
$\log_2$(Fold change)
$\log_2$(Average gene count)
G.2
Ontology
BP
CC
MF
1
2
3
Observed $-\log_{10}$(p)
Expected $-\log_{10}$(p)
D
Retained barcodes
Discarded barcodes
Pearson Correlation
kallisto UMI counts
H
Fold change
(0,1]
(-1,0]
(-2,-1]
(-3,-2]
(-4,-3]
Number of genes
Gene set
1
2
3

Supplementary Figure 3.16: Benchmark panel of data from Delile et al. 2019(Delile et al., 2019) (EMTAB7320). Enriched GO terms are 1-cytosolic large ribosomal subunit, 2-structural constituent of ribosome, 3-cytosolic small ribosomal subunit.

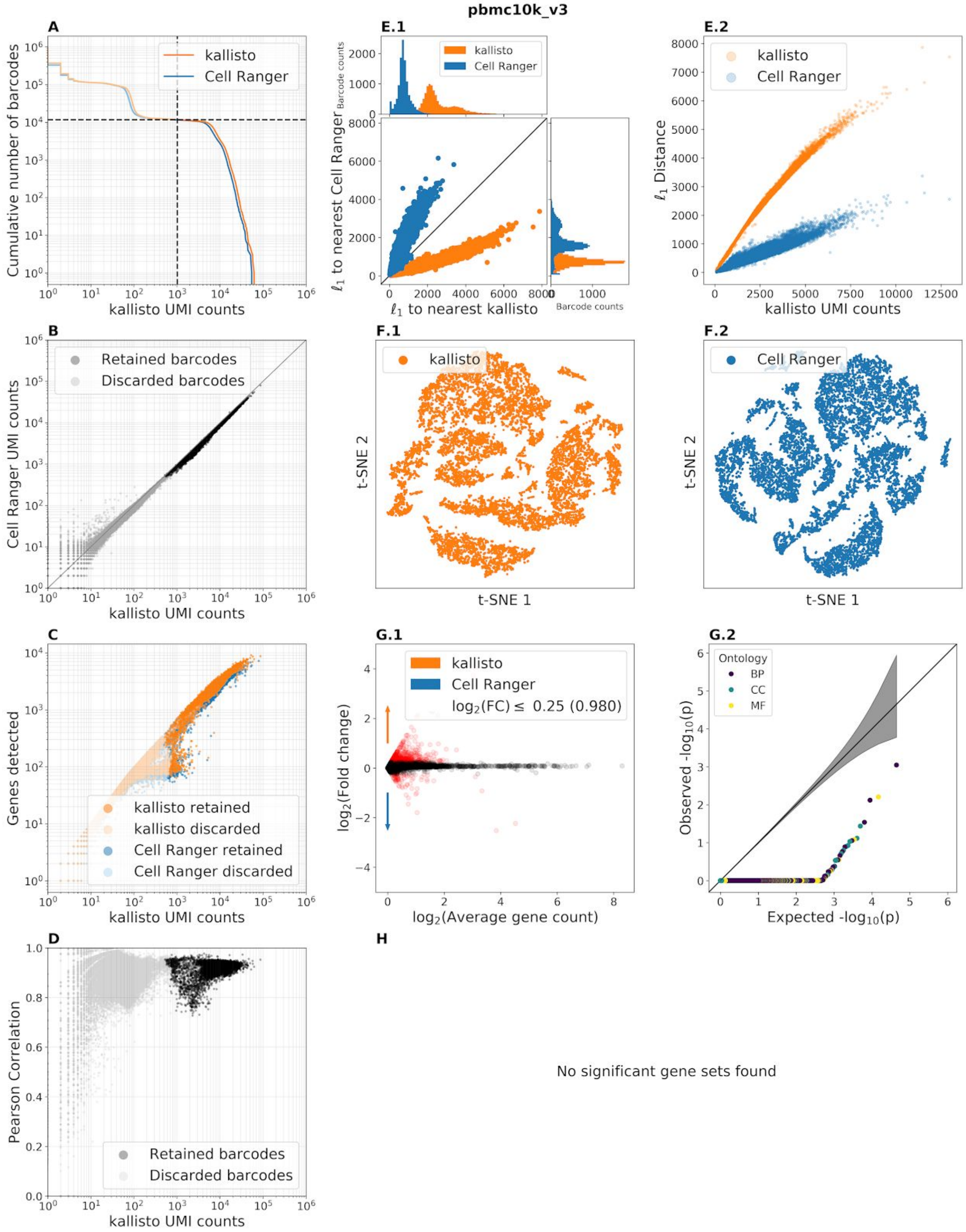


Supplementary Figure 3.17: Benchmark panel of the 10x Genomics neuron10k_v3 dataset. Enriched GO terms are 1-structural constituent of ribosome, 2-cytosolic large ribosomal subunit, 3-cytosolic small ribosomal subunit.

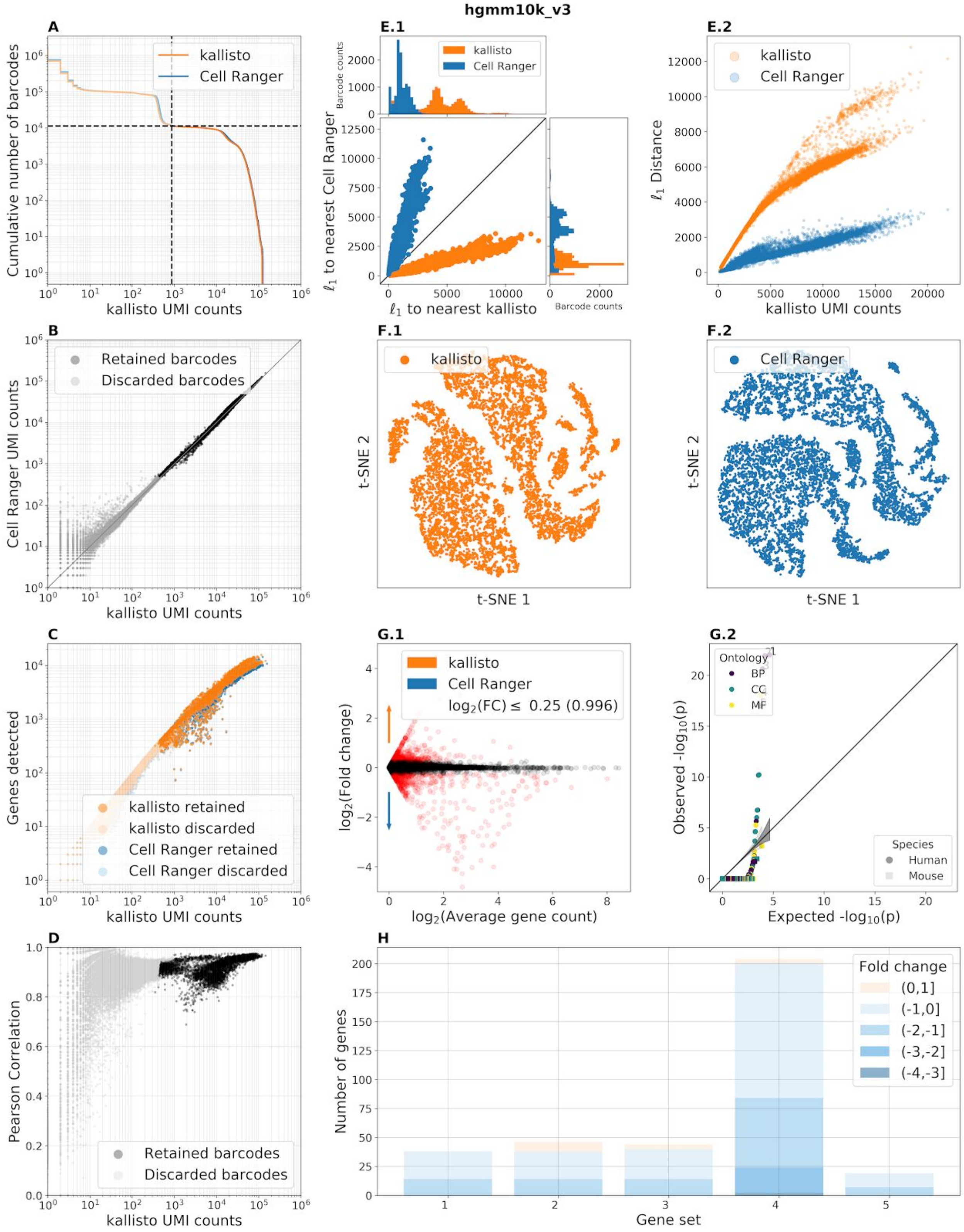


Supplementary Figure 3.18: Benchmark panel of data from Guo et al. 2019 (Guo et al., 2019) (SRR8639063). Note that the FASTQ files distributed with this experiment contained only retained barcodes. Enriched GO terms are 1-structural constituent of ribosome, 2-

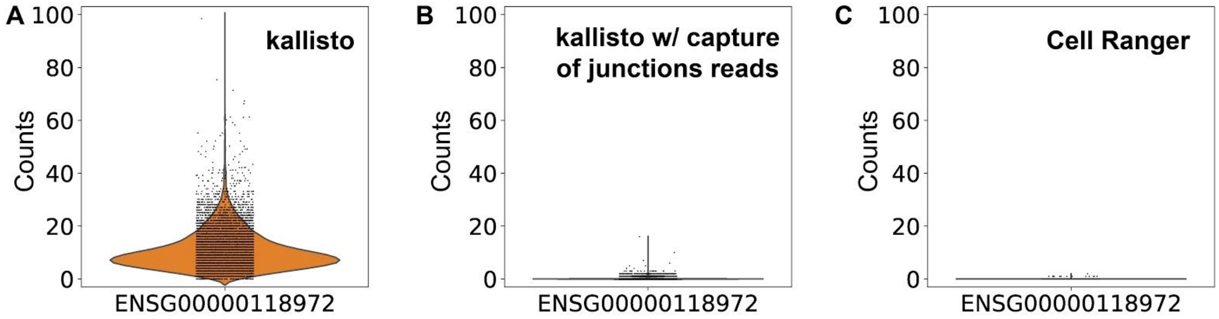
cytosolic large ribosomal subunit, 3-cytosolic small ribosomal subunit, 4-cytoplasmic translation, 5-polysomal ribosome.



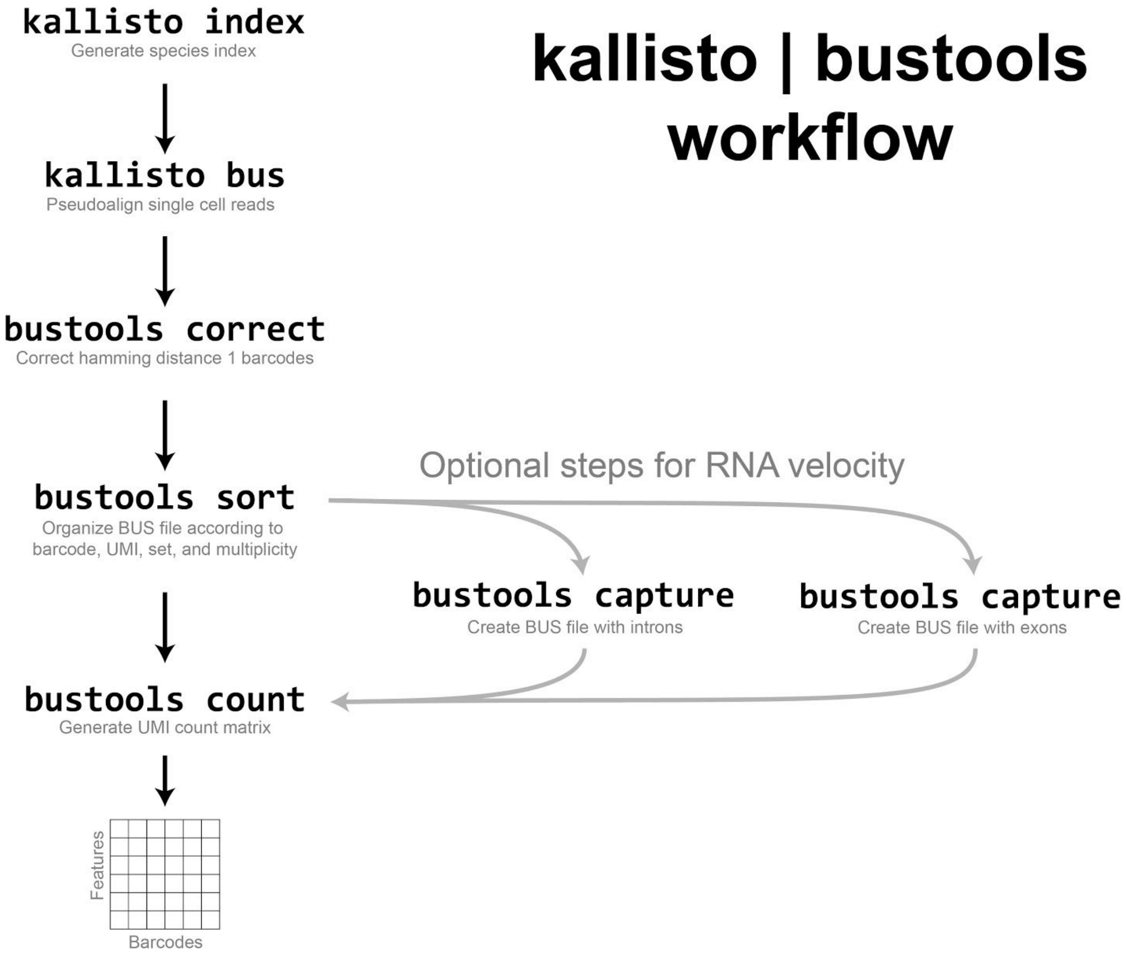
Supplementary Figure 3.19: Benchmark panel of the 10x Genomics pbmc10k_v3 dataset.



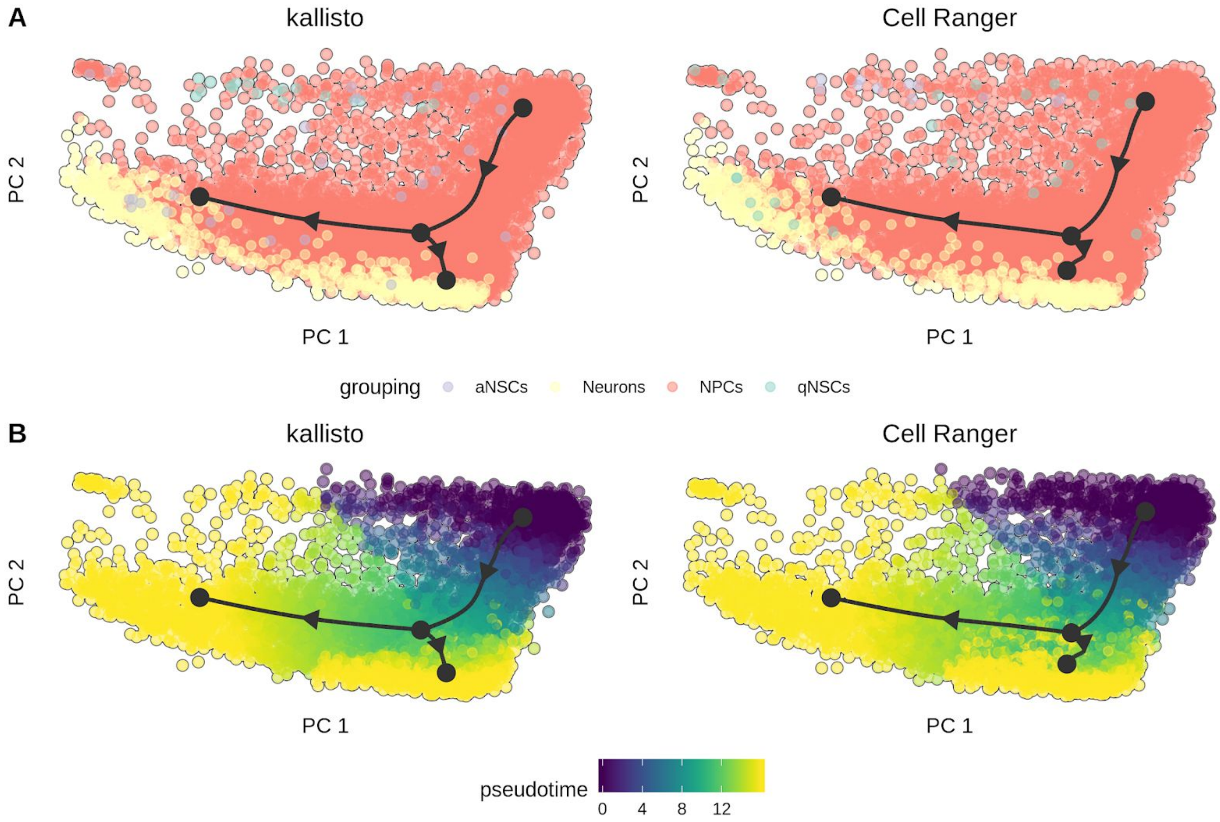
Supplementary Figure 3.20: Benchmark panel of the 10x Genomics hgmm10k_v3 dataset. Enriched GO terms are 1-SRP-dependent cotranslational protein targeting to membrane, 2-nuclear-transcribed mRNA catabolic process, nonsense-mediated decay, 3-translational initiation, 4-structural constituent of ribosome, 5-viral transcription.



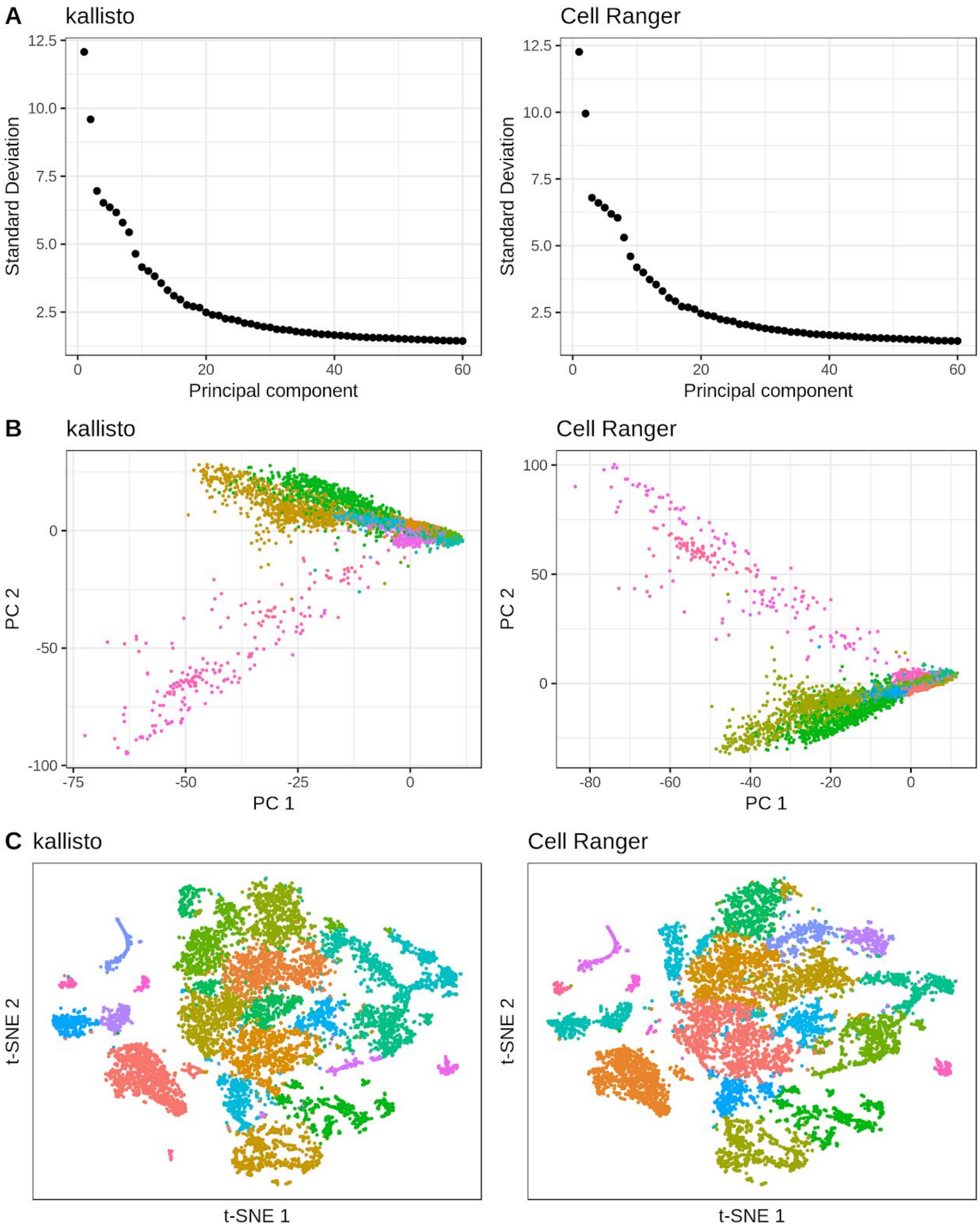
Supplementary Figure 4: Violin plots displaying distribution of counts for the FGF23 (ENSG00000118972) gene in all cells in the pbmc_10k_v3 dataset using different alignment methods. (A) Transcriptome pseudoalignment with kallisto using a standard index constructed from ENSEMBL transcripts. (B) Transcriptome pseudoalignment with kallisto using a modified index that includes, separately, sequences from splice junctions to capture unspliced junction reads. (C) Genome alignment with Cell Ranger. The gene was selected as an example as it was an outlier in discrepancy between kallisto and Cell Ranger when quantification was done with the standard index.


kallisto | bustools
workflow
kallisto index
Generate species index
kallisto bus
Pseudoalign single cell reads
bustools correct
Correct hamming distance 1 barcodes
bustools sort
Organize BUS file according to barcode, UMI, set, and multiplicity
Optional steps for RNA velocity
bustools capture
Create BUS file with introns
bustools capture
Create BUS file with exons
bustools count
Generate UMI count matrix
Features
Barcodes

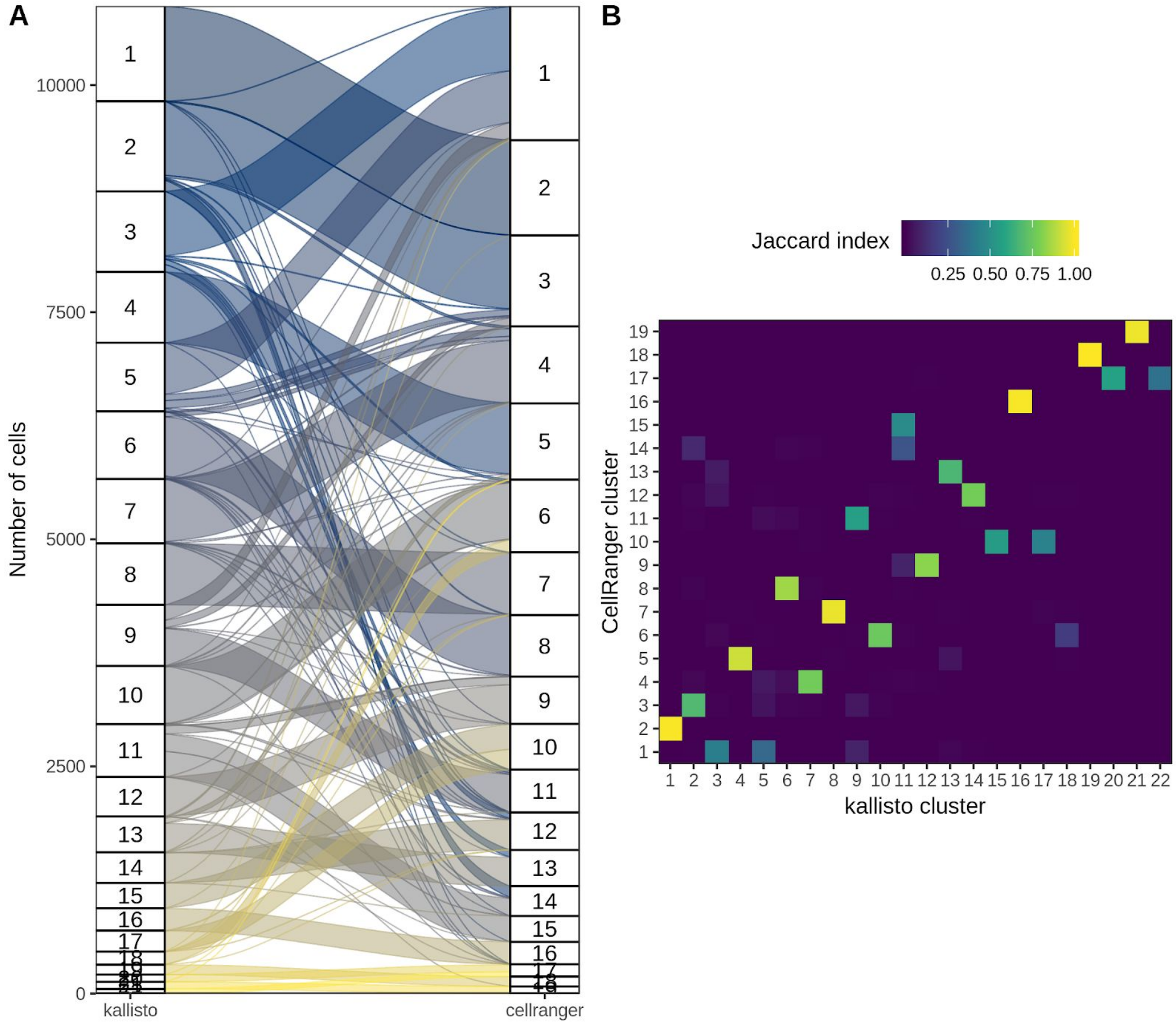
Supplementary Figure 5: Overview of the kallisto | bustools workflow. First an index for kallisto is built from a set of transcript sequences using the **kallisto index** command. Then **kallisto bus** is run on the FASTQ files; this generates a BUS file that contains records corresponding to reads, with data on the cell barcode, UMI, and transcript compatibility of each read. The barcodes are then corrected by processing the BUS file with the **bustools correct** command, after which the BUS file is sorted with **bustools sort**. Here, duplicate reads (those reads sharing an identical cell barcode, UMI, and equivalence class triplet) are collapsed into a single record and their abundance saved as a new metadata column in the BUS file named “multiplicity”. Finally, **bustoools count** produces *cells x features* count matrices. If **kallisto bus** is run with an index containing intron sequences, the **bustools capture** command can be used to produce spliced and unspliced matrices for RNA velocity after sorting and before counting.



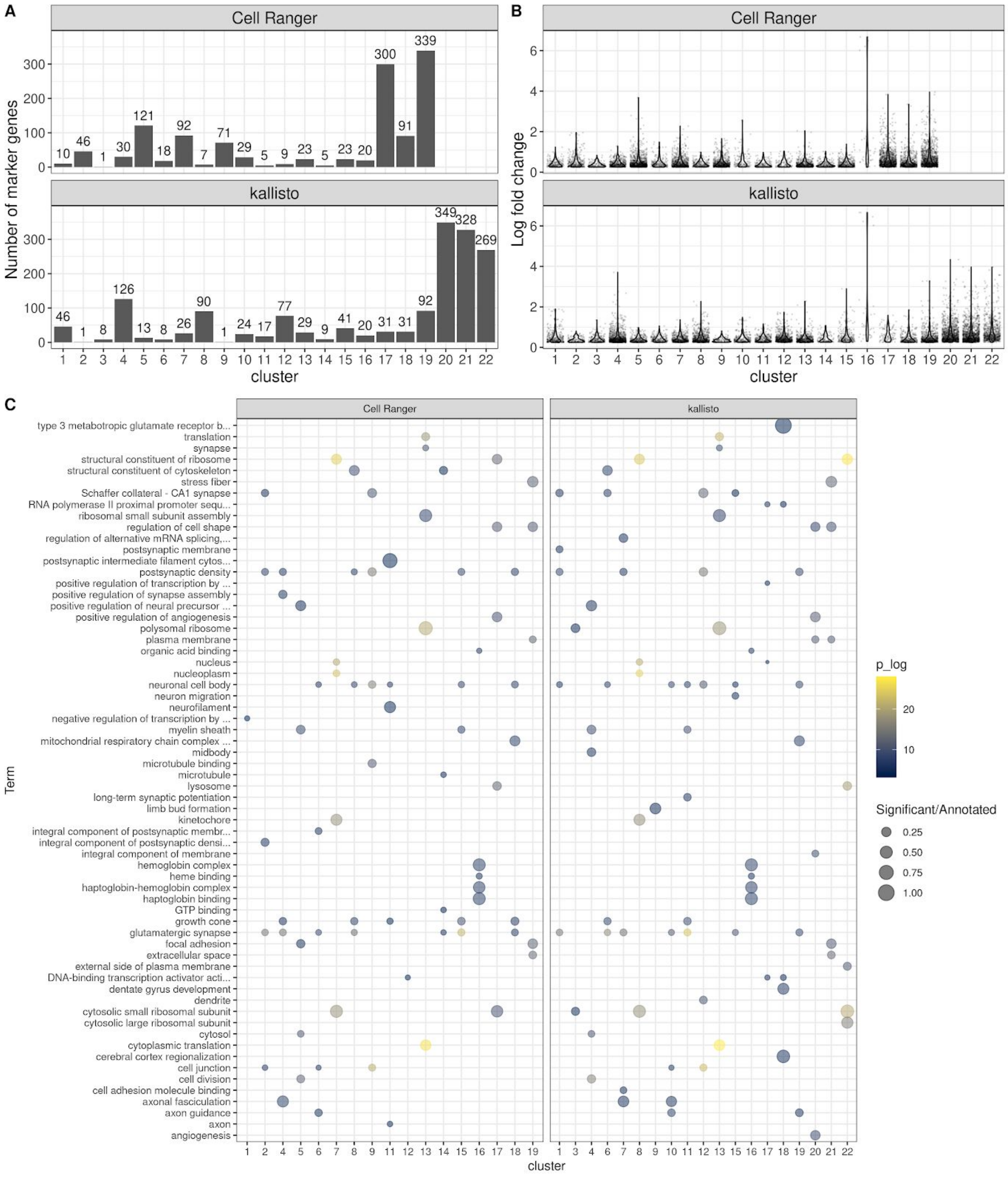
Supplementary Figure 6: Pseudotime trajectories. (A) Lineage inference of neuron10k_v3 dataset with slingshot projected to the first 2 principal components, with cells colored by cell type inferred by SingleR. aNSCs stands for active neuronal stem cells. NPCs stands for neuronal precursor cells. qNSCs stands for quiescent neuronal stem cells. (B) Coloring by pseudotime values from slingshot.



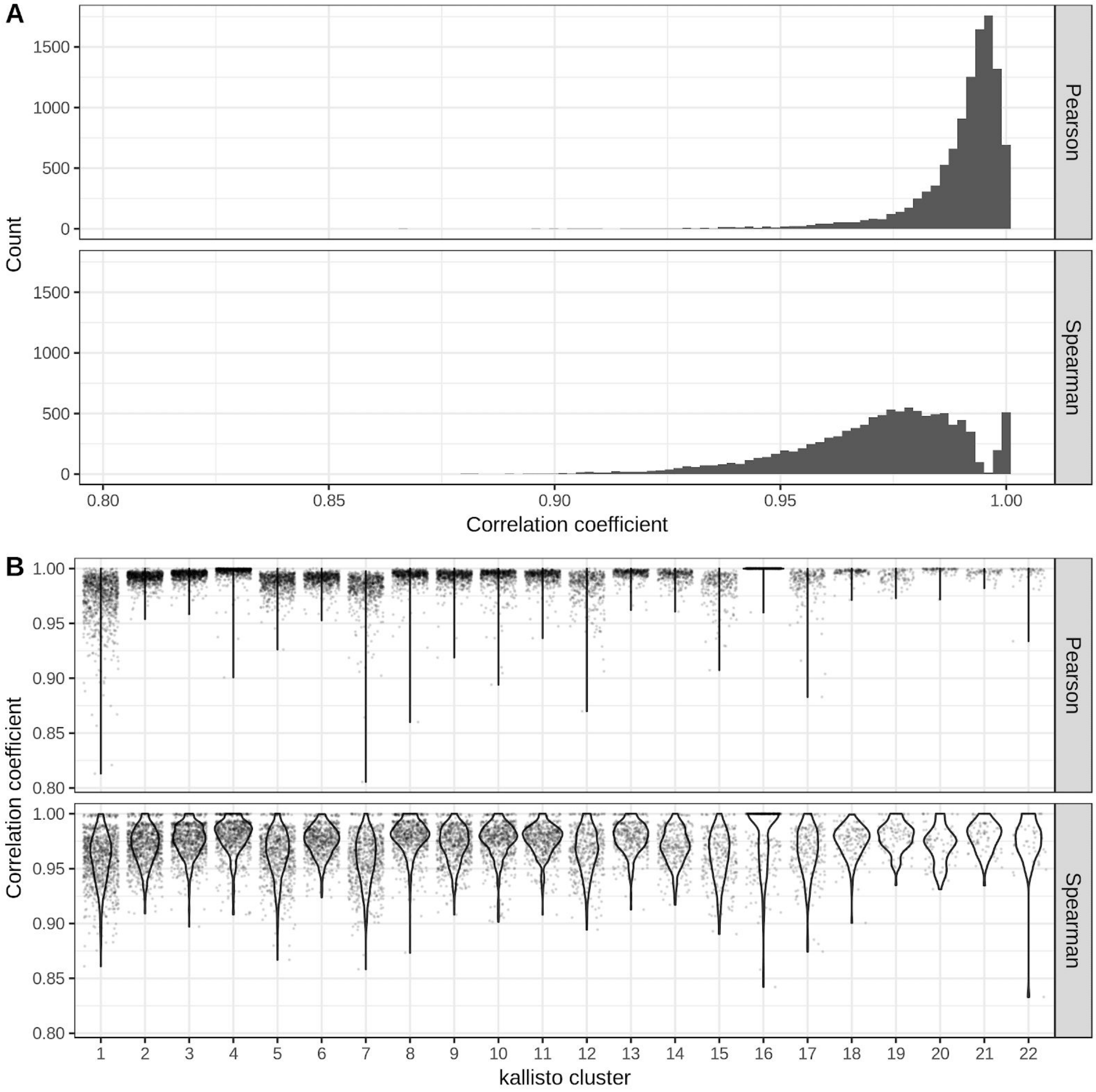
Supplementary Figure 7.1: (A) Elbow plot of standard deviation explained by each principal component of the gene count matrix from kallisto and Cell Ranger. (B) Cell embedding in the first 2 principal components colored by cluster. (C) Cell embedding in tSNE colored by cluster.



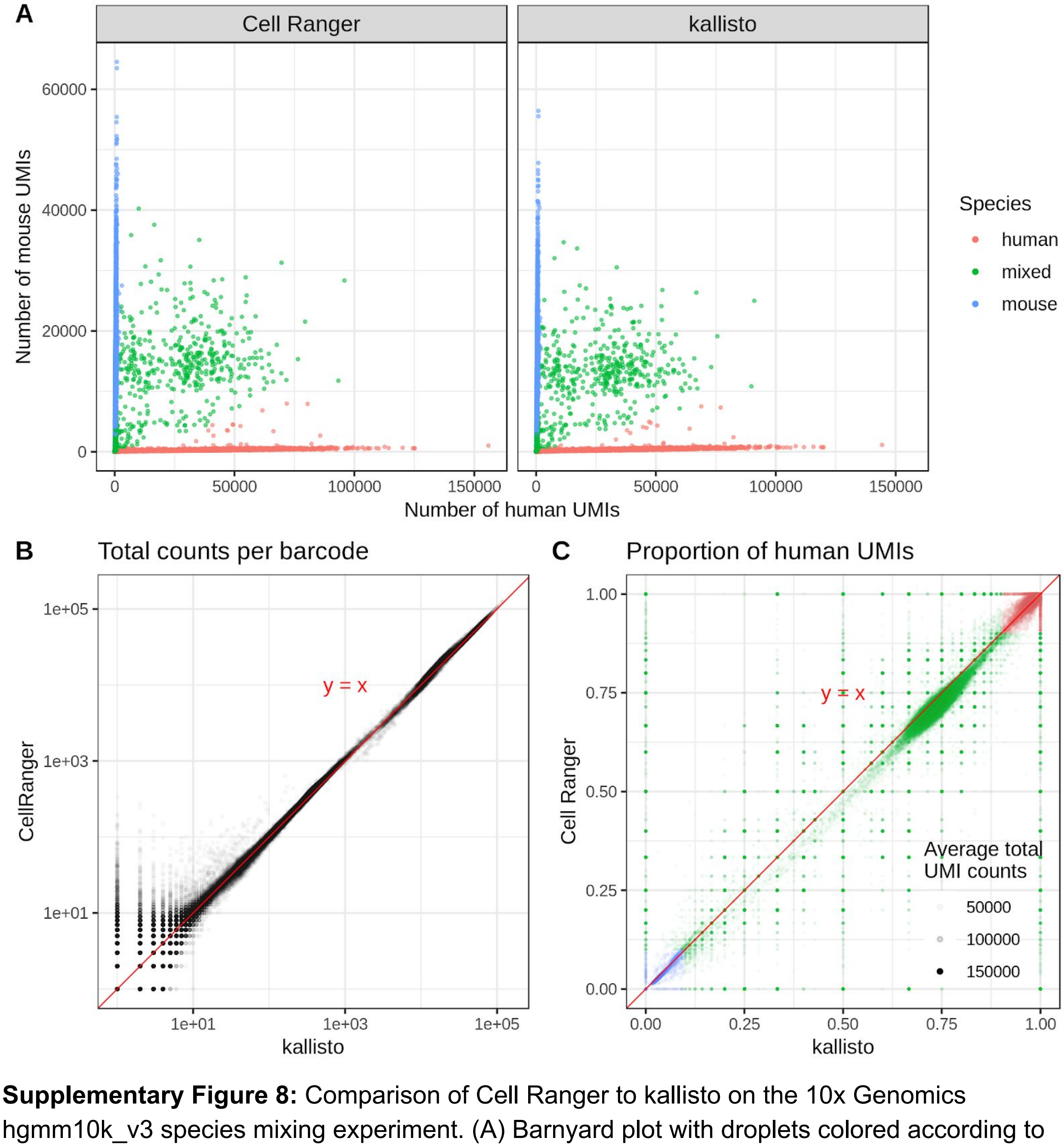
Supplementary Figure 7.2: (A) Number of cells assigned to each cluster by kallisto and Cell Ranger and the correspondence between the clusters. (B) Jaccard indices between each kallisto cluster and each Cell Ranger cluster.



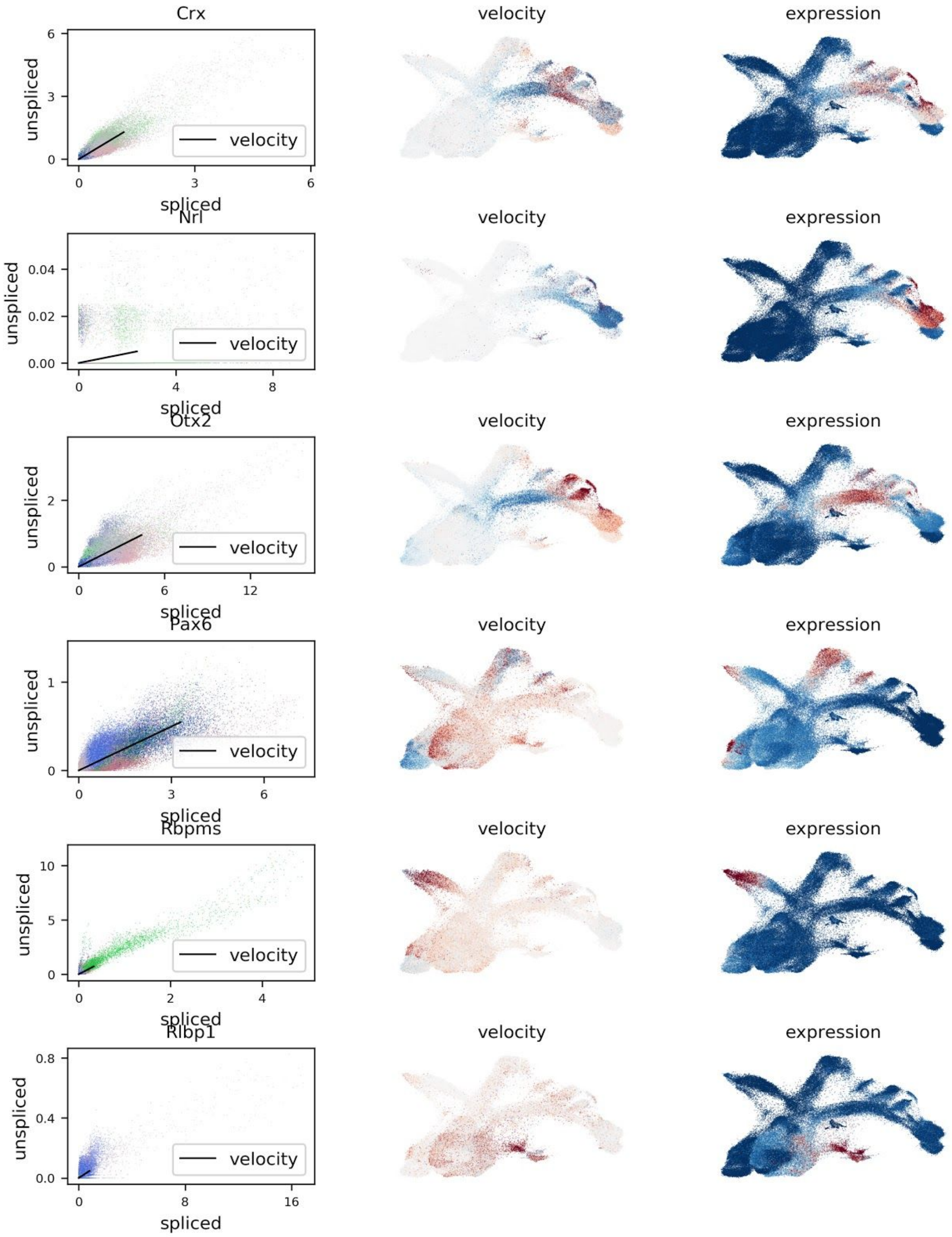
Supplementary Figure 7.3: Number of marker genes with log fold change of at least 0.75 and adjusted $p < 0.05$ in each cluster. (B) Log fold change of marker genes in each cluster. (C) Top 5 enriched GO terms of marker genes (adjusted $p < 0.05$) each cluster.



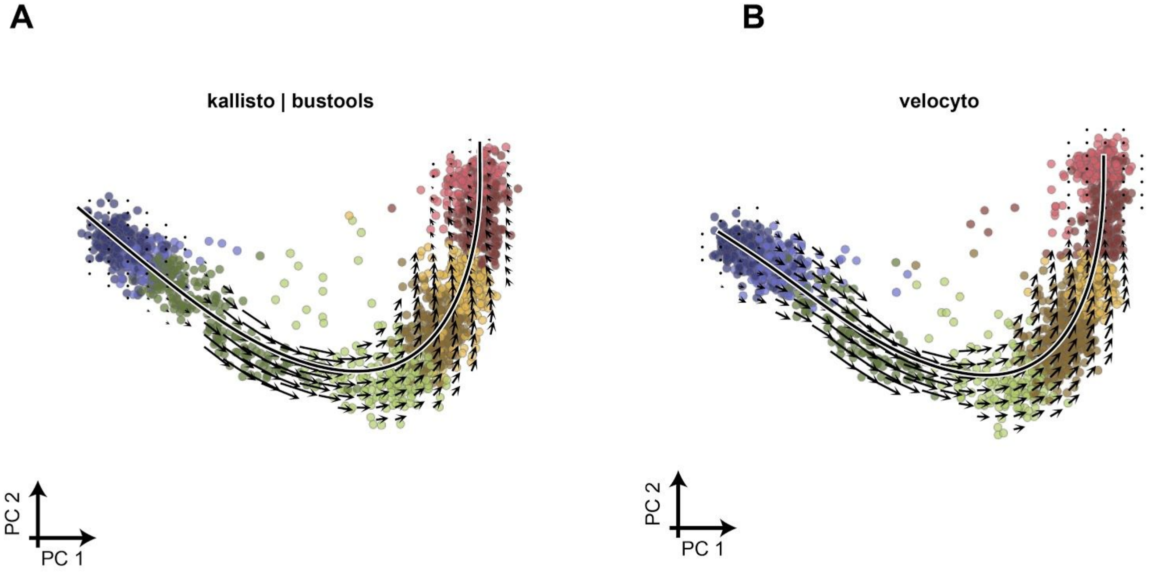
Supplementary Figure 7.4: (A) Histograms of Spearman and Pearson correlation coefficients between barcodes from kallisto and the same barcodes from Cell Ranger for the top 15 marker genes (by log fold change) of each cluster. (B) Spearman and Pearson correlation coefficients, as in (A), for cells in each kallisto cluster. Here cluster 16 corresponds to erythrocytes, while most other cells are neuronal precursor cells.



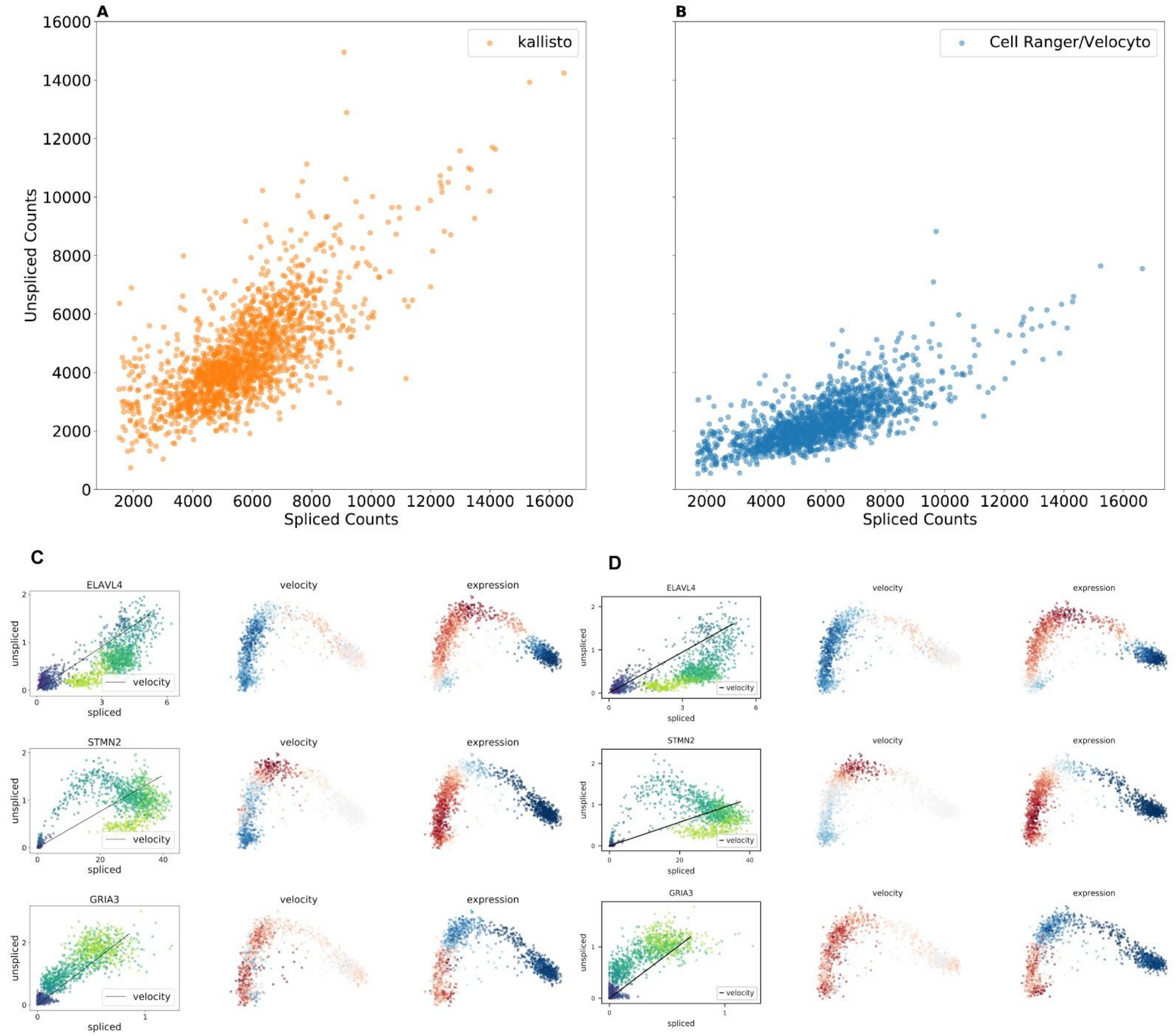
Supplementary Figure 8: Comparison of Cell Ranger to kallisto on the 10x Genomics hgmm10k_v3 species mixing experiment. (A) Barnyard plot with droplets colored according to species of origin: human (red), mouse (blue) and mixed (green). Mixed droplets correspond to cell doublets. (B) The number of total counts per barcode in Cell Ranger and kallisto. (C) The proportion of UMIs in each droplet originating from human. The cluster of droplets in the lower left corner correspond to mouse cells. The cluster of cells in the upper right corner to human cells. The middle band of droplets are doublets. Droplets are shaded according to the number of distinct UMIs they contain.



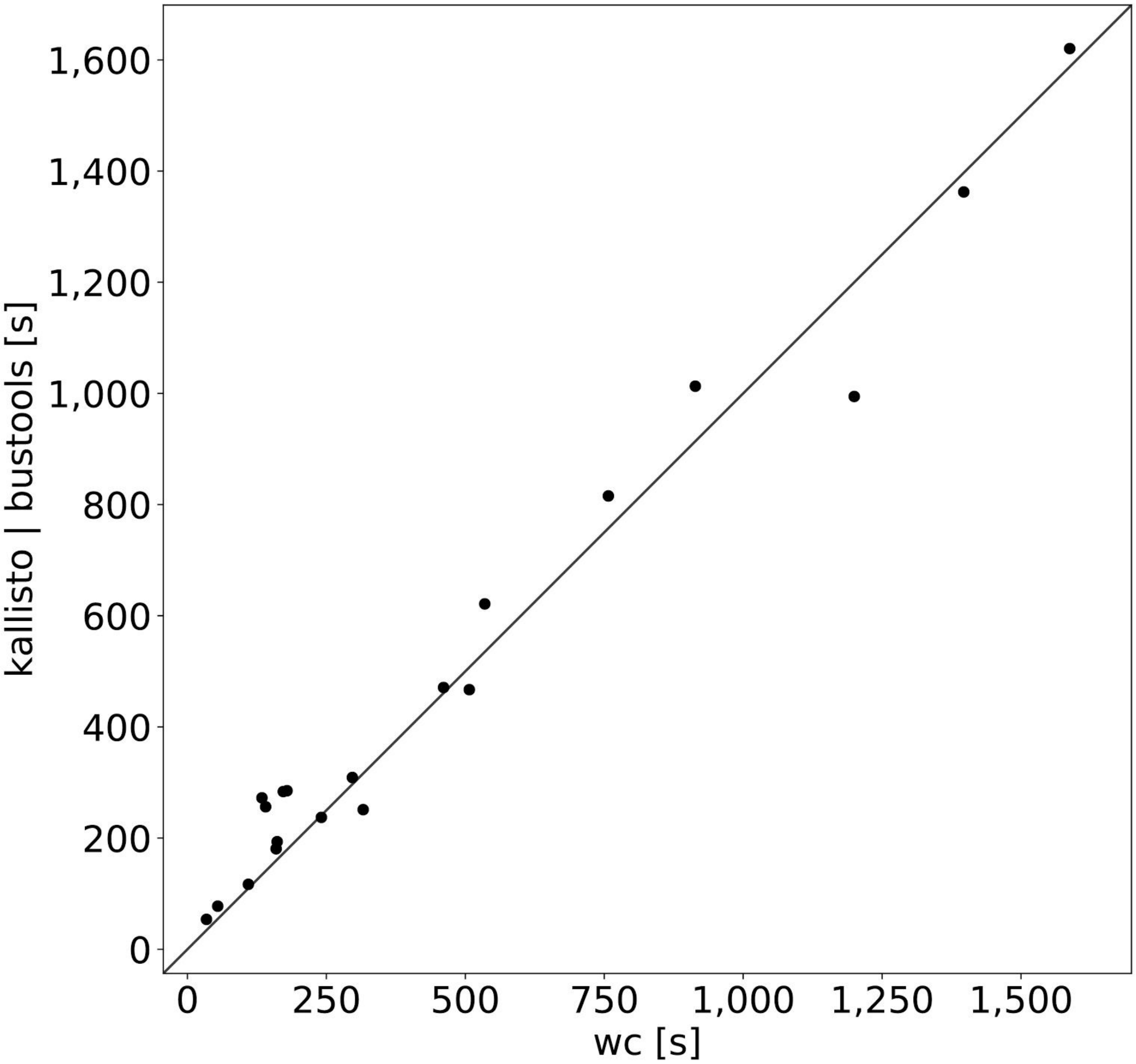
Supplementary Figure 9: Phase diagrams and expression/velocity for six marker genes studied in Clark et al. 2019. The expression results are concordant with pseudotime analysis.



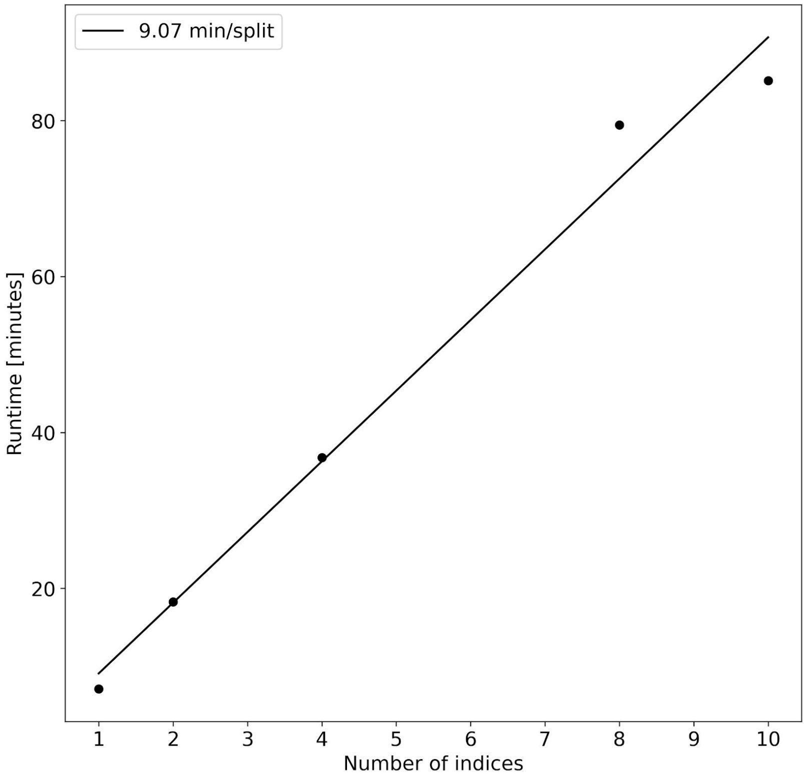
Supplementary Figure 10: (A) RNA velocity based on spliced and unspliced matrices from a dataset of 1,720 human glutamatergic neuron differentiation cells at post-conception week 10. The colors correspond to cell types and intermediate states and a principal “velocity curve” is shown in bold. (A) RNA velocity analysis based on spliced and unspliced matrices computed with kallisto and bustools. B) RNA velocity based on the spliced and unspliced matrices computed with velocyto. Colors correspond to clusters as assigned by the velocyto notebook.



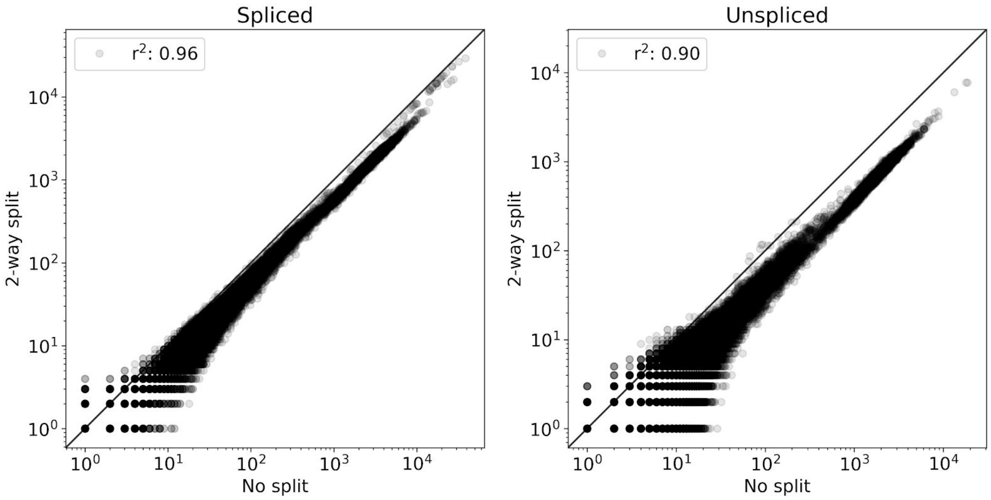
Supplementary Figure 11: Comparison of Cell Ranger and velocyto to kallisto in an RNA velocity analysis of human glutamatergic neuron differentiation cells at post-conception week 10. (A) Number of distinct UMIs from spliced vs. unspliced transcripts from kallisto (orange). (B) Number of distinct UMIs from spliced vs. unspliced transcripts from Cell Ranger (blue). Cell Ranger has similar numbers of spliced counts but fewer unspliced counts. (C) Phase diagrams from the kallisto RNA velocity analysis for 3 genes highlighted in La Manno et al. 2018. (D) Corresponding phase diagrams from the Cell Ranger RNA velocity analysis showing agreement with the kallisto results.



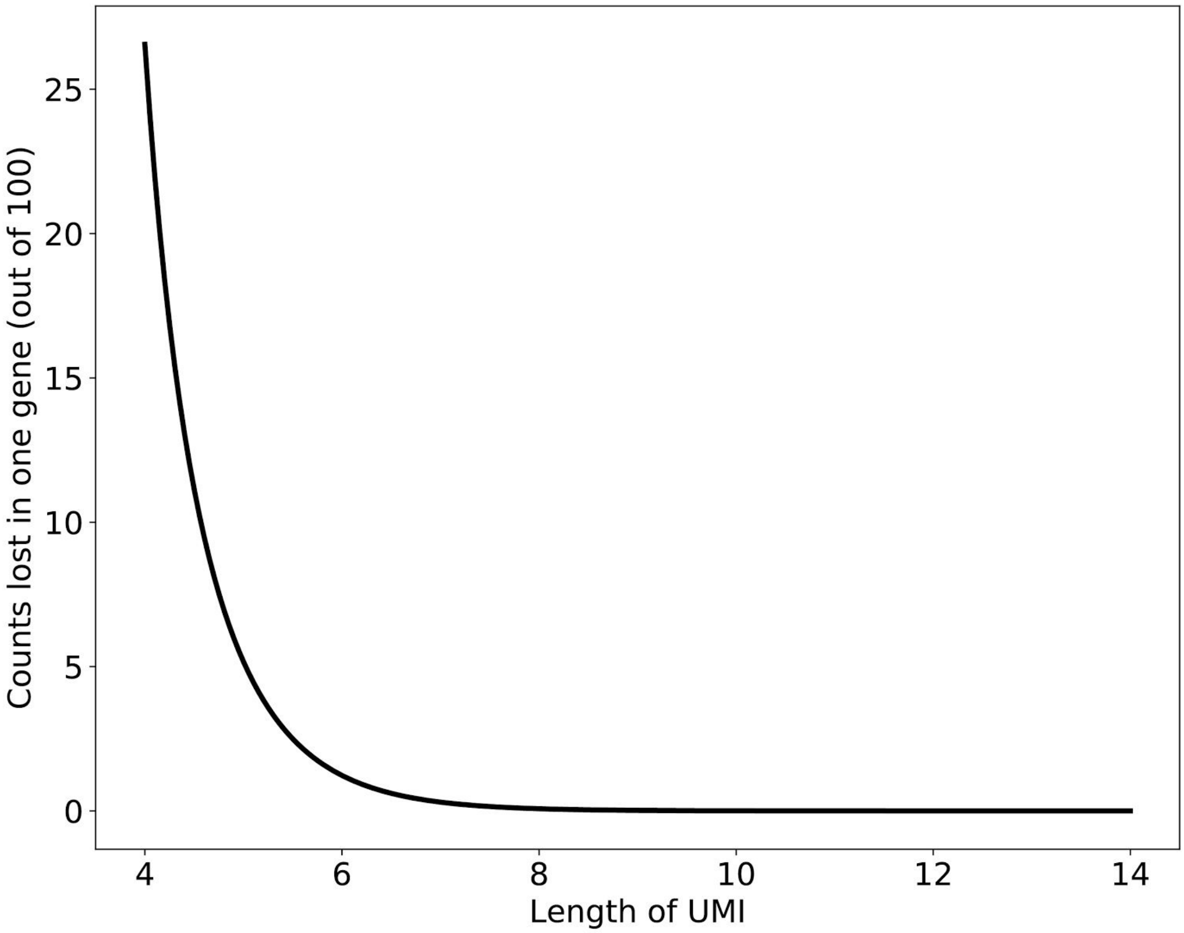
Supplementary Figure 12: Comparison of kallisto runtimes with those of the Unix word count (**wc**) command. Each point corresponds to a different dataset.



Supplementary Figure 13: The runtime to process 50 million reads as a function of the number of indices. The reference transcriptome was split into two, four, eight, and ten parts and the time to align all of the reads to each of the set of indices was recorded.



Supplementary Figure 14: The counts per cell, summed across all genes, when pseudoaligning 50 million single cell reads against the full spliced and unspliced indices and the 2-way split index for spliced and unspliced count matrices. The BUS files generated for the 2-way split index were merged together using bustools mash followed by bustools sort and bustools merge.



Supplementary Figure 15: The number of counts lost due to naïve collapsing of UMIs as a function of the length of the UMIs for a gene with 100 counts. The calculation, based on Supplementary Note equation (11), assumes that the effective number of UMIs is 4^L when UMIs are of length L.

Supplementary Table 1: Runtime, memory, and cost (supplementary_table_S1_runtime_mem_cost.xlsx).

bustools	Description	Enables
capture	Capture records from a BUS file	RNA Velocity
correct	Error correct a BUS file	Barcode Error correction
count	Generate count matrices from a BUS file	Gene count or transcript count matrices
extract	Extract FASTQ reads corresponding to reads in BUS file	FASTQ sampling
inspect	Produce a report summarizing a BUS file	Summary statistics
linker	Remove section of barcodes in BUS files	Excise sections of barcode for custom technologies
mash	Combine BUS records and match EC to the same reference	Combining BUS files from different indices
merge	Merge kmer alignments for a single read	Low memory alignment
project	Project a BUS file to gene sets	Change coordinate system from transcripts to genes
sort	Sort a BUS file by barcodes and UMIs	Constant memory sorting
text	Convert a binary BUS file to a tab-delimited text file	Custom BUS file parsing
whitelist	Generate a whitelist from a BUS file	Technologies without a whitelist

Supplementary Table 2: All of the bustools commands that have been developed and the types of analyses they enable.

Supplementary Table 3: Benchmark panel summary (supplementary_table_S3 _benchmark_panel_summary.xlsx).

Bibliography

Carosso, G.A., Boukas, L., Augustin, J.J., Nguyen, H.N., Winer, B.L., Cannon, G.H., Robertson, J.D., Zhang, L., Hansen, K.D., Goff, L.A., et al. (2018). Transcriptional suppression from KMT2D loss disrupts cell cycle and hypoxic responses in neurodevelopmental models of Kabuki syndrome: BioRxiv.

Delile, J., Rayon, T., Melchionda, M., Edwards, A., Briscoe, J., and Sagner, A. (2019). Single cell transcriptomics reveals spatial and temporal dynamics of gene expression in the developing mouse spinal cord. Development *146*.

Farrell, J.A., Wang, Y., Riesenfeld, S.J., Shekhar, K., Regev, A., and Schier, A.F. (2018). Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. Science *360*.

Guo, L., Lin, L., Wang, X., Gao, M., Cao, S., Mai, Y., Wu, F., Kuang, J., Liu, H., Yang, J., et al. (2019). Resolving Cell Fate Decisions during Somatic Cell Reprogramming by Single-Cell RNA-Seq. Mol. Cell *73*, 815-829.e7.

Jin, R.M., Warunek, J., and Wohlfert, E.A. (2018). Chronic infection stunts macrophage heterogeneity and disrupts immune-mediated myogenesis. JCI Insight *3*.

Mahadevaraju, S., Fear, J.M., and Oliver, B. (2019). GEO Accession viewer.

Mays, J.C., Kelly, M.C., Coon, S.L., Holtzclaw, L., Rath, M.F., Kelley, M.W., and Klein, D.C. (2018). Single-cell RNA sequencing of the mammalian pineal gland identifies two pinealocyte subtypes and cell type-specific daily patterns of gene expression. PLoS ONE *13*, e0205883.

Merino, D., Weber, T.S., Serrano, A., Vaillant, F., Liu, K., Pal, B., Di Stefano, L., Schreuder, J., Lin, D., Chen, Y., et al. (2019). Barcoding reveals complex clonal behavior in patient-derived xenografts of metastatic triple negative breast cancer. Nat. Commun. *10*, 766.

Miller, B.C., Sen, D.R., Al Abosy, R., Bi, K., Virkud, Y.V., LaFleur, M.W., Yates, K.B., Lako, A., Felt, K., Naik, G.S., et al. (2019). Subsets of exhausted CD8+ T cells differentially mediate tumor control and respond to checkpoint blockade. Nat. Immunol. *20*, 326–336.

O’Koren, E.G., Yu, C., Klingeborn, M., Wong, A.Y.W., Prigge, C.L., Mathew, R., Kalnitsky, J., Msallam, R.A., Silvin, A., Kay, J.N., et al. (2019). Microglial Function Is Distinct in Different Anatomical Locations during Retinal Homeostasis and Degeneration. Immunity *50*, 723-737.e7.

Packer, J.S., Zhu, Q., Huynh, C., Sivaramakrishnan, P., Preston, E., Dueck, H., Stefanik, D., Tan, K., Trapnell, C., Kim, J., et al. (2019). A lineage-resolved molecular atlas of C. elegans embryogenesis at single cell resolution. BioRxiv.

Ryu, K.H., Huang, L., Kang, H.M., and Schiefelbein, J. (2019). Single-Cell RNA Sequencing Resolves Molecular Relationships Among Individual Plant Cells. Plant Physiol. *179*, 1444–1456.

Supplementary Note

Modular and efficient pre-processing of single-cell RNA-seq

Páll Melsted*, A. Sina Booeshaghi*, Lauren Liu, Fan Gao, Lambda Lu, Kyung Hoi (Joseph) Min,
Eduardo da Veiga Beltrame, Kristján Eldjárn Hjörleifsson, Jase Gehring, and Lior Pachter

Address correspondence to lpachter@caltech.edu
* Authors contributed equally

1 Preliminaries

$$\begin{tikzcd}
 & L & F \arrow[l, tail] & B \arrow[l, tail] \\
R & C \arrow[l, tail] \arrow[u, hook] \arrow[dl, tail] & & T \arrow[ll, tail] \arrow[u, tail] \\
U & & I \arrow[ll, tail] \arrow[ur, leftrightarrow] &
\end{tikzcd}$$

Figure 1: Diagram of sets associated with a cell in a single-cell RNA-seq sequencing experiment.

A single-cell RNA-seq experiment can be described as follows: the goal of the experiment is to identify the ensemble of RNA molecules in multiple cells; in Figure 1 the ensemble of RNA molecules contained within a single cell is denoted by R. To investigate R a library (L) is constructed from the set of molecules captured from R (the set C). Typically, L is the result of of various fragmentation and amplification steps performed on C, meaning each element of C may be observed in L with some multiplicity. Thus, there is an inclusion map from C to L, and an injection from C to R. The library is interrogated via sequencing of some of the molecules in L, resulting in a set F of fragments. Subsequently, the set F is aligned or pseudoaligned to create a set B, which in this paper is a BUS file (Melsted, Ntranos, and Pachter 2019). Not every fragment F is represented in B, hence the injection, rather than bijection, from

R : multiset of all RNAs in a cell.
L : multiset of all molecules in the library.
F : multiset of reads.
B : multiset of {barcode, UMI, equivalence class} triplets.
C : multiset of captured RNA molecules represented in the library.
T : multiset of transcripts represented in B corresponding to molecules in C.
U : set of UMIs on a bead.
I : multiset of UMIs represented in B corresponding to molecules in U.

Table 1: Notation for description of a single-cell experiment.

B to F, and similarly from F to L. The set T consists of transcripts that correspond to molecules in C that were represented in B. Note that $|R| \geq |C| \geq |T|$. Separately, the set U consists of the UMIs on the bead that the cell was trapped with, and I is a multiset of UMIs associated with transcripts in C and UMIs in U that are in B (Table 1).

The data in a single-cell experiment consists of the sets F for each cell. In our workflow, a combined BUS file (merge of the sets B) is generated using kallisto (Bray et al. 2016). While the multiset I is not directly measured, it's support $supp(I)$ (the set of distinct UMIs) can be extracted from the BUS file. The goal of single-cell RNA-seq pre-processing is to infer the multiset T. What we describe in this note is an approach to estimating two different quantities: the effective size $|U|$ of the set of UMIs associated with each bead, and the number of captured molecules represented in the BUS file, i.e. $|I|$ or equivalently $|T|$. Specifically, we are interested in the restriction of the latter to individual genes in cells, for the purpose of estimating the error in the number of counts that can be introduced when naïvely collapsing UMIs by gene. The reason for estimating $|U|$ is that it is necessary to estimate $|I|$.

2 Modeling an experiment

The number of distinct UMIs on one bead is at most 4^L where L is the number of UMI bases (10xv2 technology uses $L = 10$ and 10xv3 technology $L = 12$). For a bead captured along with a cell in a droplet, we denote the number of UMIs on the bead by $n = |U|$. We model the process by which UMIs are associated with molecules as follows: each UMI is selected by sampling uniformly at random from the set of UMIs U. In other words, the molecules are labeled with UMIs by sampling with replacement. This model has been used previously (Grün, Kester, and Oudenaarden 2014), and is justified by distributions of UMIs seen empirically (Figure 2). If $k = |I|$ is the number of UMIs represented in B corresponding to molecules in U derived from a single droplet then the assumption of uniform random sampling of UMIs from the bead implies that the probability that a specific UMI is observed zero times is $\left(1 - \frac{1}{n}\right)^k$. Therefore

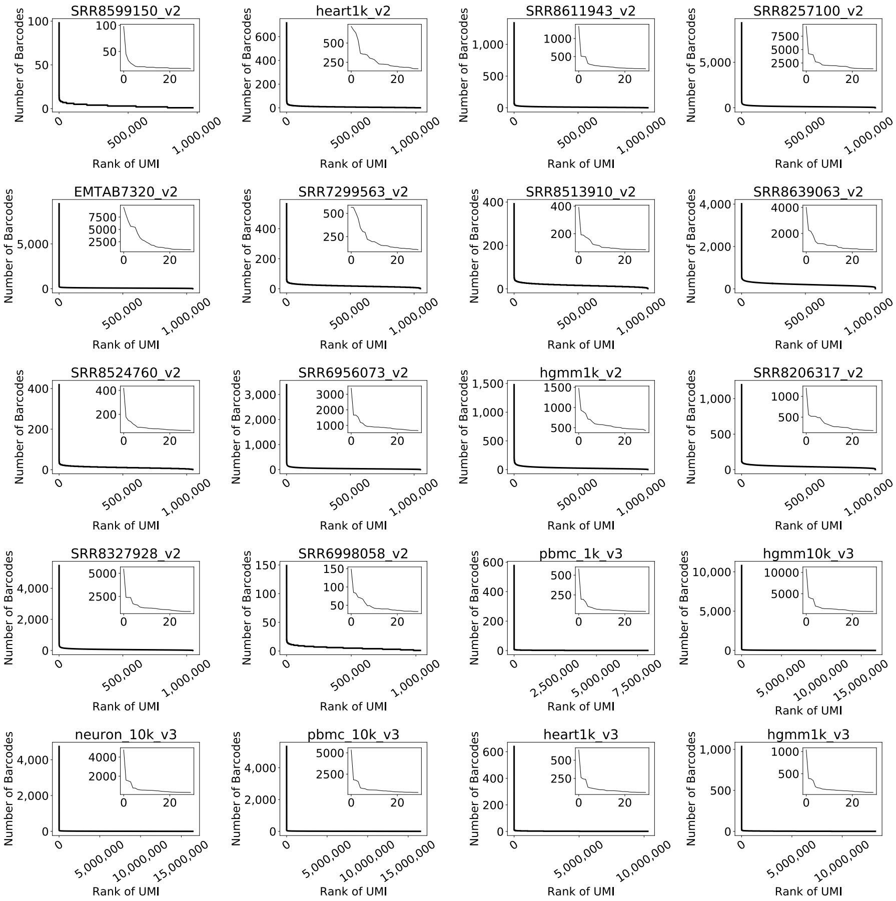


Figure 2: Distribution of UMIs across cells. With the exception of a handful of artifacts, UMIs are uniformly distributed across cells.

the expected number of UMIs observed at least once, i.e. the expected number of distinct UMIs in a cell, is

$$n\left(1-\left(1-\frac{1}{n}\right)^{k}\right). \tag{1}$$

3 Estimating the effective number of UMIs

To estimate n ($=|U|$) we utilize two observations:

1. Reads that originated from different genes correspond to distinct molecules, so if they share the same UMI then the UMI was sampled more than once (i.e. the UMI is not duplicated due to PCR).

2. While the number of sampled UMIs k is unknown, the number of distinct UMIs can be measured directly.

We say that a UMI that has been sampled more than once has a *collision* (Figure 3), and we denote the number of UMIs that appear in more than one gene by random variable (r.v.) X (see Figure 3). We denote the number of distinct UMIs sequenced, i.e. $|supp(I)|$, by a r.v. D, and the number of distinct UMIs observed to be from a gene g by r.v. D_g. We denote the number of sampled molecules originating from a gene g by k_g. Note that $\sum_g \mathbb{E}[d_g] \geq \mathbb{E}[d]$.

We obtain method of moment estimates for the parameters k, k_g and n ($\hat{k}, \hat{k_g}$ and $\hat{n}$) by relating them to realizations of the r.v. D_g, D, and X. First, from equation (1) (see also Grün, Kester, and Oudenaarden 2014), we have that

$$\mathbb{E}[D] = n\left(1 - \left(1 - \frac{1}{n}\right)^k\right), \tag{2}$$

and at the gene level,

$$\mathbb{E}[D_g] = n\left(1 - \left(1 - \frac{1}{n}\right)^{k_g}\right). \tag{3}$$

The number of UMIs that occur in more than one gene can be found by knowing the number of UMIs that are seen zero times in all genes, and that the number of UMIs that are seen in only one gene is given by the number of unique UMIs in gene g and only gene g, summed across all genes. This gives an estimate for the number of UMIs that collide between genes:

$$\mathbb{E}[X] = n\left(1 - \left(\left(1 - \frac{1}{n}\right)^k + \sum_g \left(1 - \left(1 - \frac{1}{n}\right)^{k_g}\right) \cdot \left(1 - \frac{1}{n}\right)^{k-k_g}\right)\right). \tag{4}$$

From equations (2) and (3) and using the realizations of the r.v. D and D_g, i.e. d and d_g, we have that

$$\left(1 - \frac{1}{\hat{n}}\right)^{\hat{k}} = \frac{\hat{n} - d}{\hat{n}} \tag{5}$$

and at the gene level

$$\left(1 - \frac{1}{\hat{n}}\right)^{\hat{k_g}} = \frac{\hat{n} - d_g}{\hat{n}}. \tag{6}$$

Therefore, substituting equations (5), (6) into equation (4) we obtain

$$\begin{aligned}
x &= \hat{n}\left(1 - \left(\frac{\hat{n} - d}{\hat{n}} + \sum_g \left(1 - \frac{\hat{n} - d_g}{\hat{n}}\right) \cdot \left(\frac{\hat{n} - d}{\hat{n} - d_g}\right)\right)\right) &(7)\\
&= \hat{n}\left(\frac{d}{\hat{n}} - \frac{\hat{n} - d}{\hat{n}} \sum_g \left(\frac{d_g}{\hat{n} - d_g}\right)\right) &(8)\\
&= d - (\hat{n} - d) \sum_g \left(\frac{d_g}{\hat{n} - d_g}\right). &(9)
\end{aligned}$$

Since d, d_g (for all g) and x are known the number of UMIs, $\hat{n}$, can be estimated.

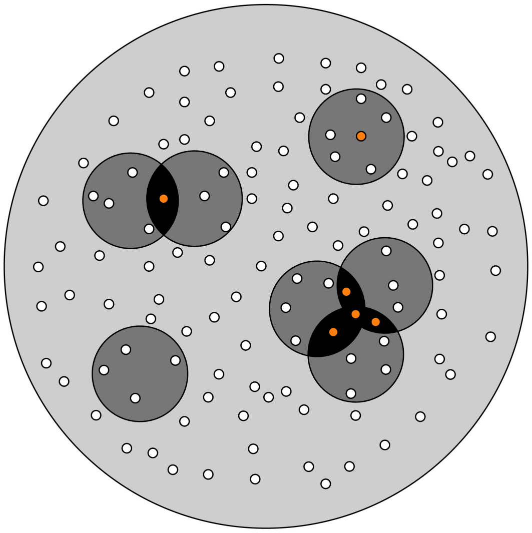


Figure 3: Collisions of UMIs. Each small circle represents a distinct UMI. Each medium sized circle is a gene, and the enclosing circle is the set of all distinct UMIs. UMIs that have collided are shown in orange. Inter-gene collisions consist of UMIs present in two or more genes. An intra-gene collision is also shown.

4 Estimating counts lost for each gene

Returning to equation (3), we see that

$$\hat{k_g} = \frac{ln\left(1 - \frac{d_g}{\hat{n}}\right)}{ln\left(1 - \frac{1}{\hat{n}}\right)}. \tag{10}$$

With $\hat{n}$, and measurement of d_g, we evaluate the number of molecules captured per gene, $\hat{k_g}$. The loss of counts due to collapsing of UMIs by gene is

$$\begin{aligned} \hat{k_g} - d_g &= \frac{ln\left(1 - \frac{d_g}{\hat{n}}\right)}{ln\left(1 - \frac{1}{\hat{n}}\right)} - d_g \qquad (11) \\ &\approx \frac{d_g(d_g - 1)}{2\hat{n} + 1}, \qquad (12) \end{aligned}$$

where (12) is found by Taylor expanding (11).

5 Constant and low memory processing

The kallisto bustools workflow enables constant and low memory single-cell RNA-seq pre-processing by using small pseudoalignment reference indices, and streaming all processing of BUS records which is possible after a constant memory sort of the initial BUS file produced in an analysis. In order to pseudoalign reads, kallisto first loads up a small index file constructed from a reference transcriptome. The size of this index is not dependent on the number of reads that will be processed. Reads are pseudoaligned by streaming through FASTQ files, and BUS records are incrementally added to as reads are processed. The bustools commands operate on BUS files and are used to perform many required operations on BUS files in order to generate count matrices. These operations include sorting the BUS file, correcting barcodes, and counting UMIs among many others; all operations are performed in constant memory in the number of reads being processed. The first step in working with BUS files is sorting. Sorting the BUS file allows all other bustools to operate on the BUS file in a stream-wise fashion thus keeping memory constant and low. The 'bustools sort' command operates in constant memory by utilizing disk when necessary.

While pseudoalignment of reads and processing of BUS files to perform RNA-velocity has only constant memory requirements (in terms of the number of reads) with the kallisto bustools workflow, the indices involved can be large due to the intronic sequences that must be indexed. The modularity of bustools makes possible, in principle, a reduction in absolute memory requirements by virtue of splitting the target sequences prior to indexing, pseudoalignment to the separate indices, and finally merging of the resultant BUS files. We implemented this strategy, which required modifying the kallisto bustools workflow to first align reads to a transcriptome that has been split into an arbitrary number (n) of parts and then merging the alignments by interval set intersection. Splitting the transcriptome into n parts yields a smaller indicies to be loaded into memory and requires n alignments of the reads which comes with a run-time trade-off (Supplementary Figure 13). For each read that aligns, we record the interval of kmer start positions from the read such that the kmers contained within this interval align to an associated equivalence class. A single read of length L, with a kmer size of k can have at most $L - k + 1$ possible kmer alignments where each possible kmer in that read aligns to a different equivalence class. We then merge these intervals appropriately in order to assign an equivalence class to the read.

By way of example, suppose that we split an index into three parts and perform pseudoalignment three times. Additionally, suppose that a single read has only two kmers that align, k_1 and k_2. k_1 aligns to an equivalence class which contains transcripts one and two ($EC_1 = \{T_1, T_2\}$) in index one and k_1 also aligns to $EC_2 = \{T_7, T_9\}$. The second kmer k_2 aligns to $EC_4 = \{T_5, T_6, T_7\}$ of index two and $EC_5 = \{T_{10}, T_{11})$ of index three.

In the case of a full transcriptome, EC_1 and EC_2 would have been indexed together since they share the same kmer, $EC_{1,2} = \{T_1, T_2, T_7, T_9\}$ and k_1 would have aligned to this equivalence class. Similarly, in the case of a full transcriptome, EC_4 and EC_5 would have have been indexed together since they share the same kmer, $EC_{4,5} = \{T_5, T_6, T_7, T_{10}, T_{11}\}$ and k_2 would have aligned to this equivalence class. The read would then have been assigned the equivalence class $EC = EC_{1,2} \cap EC_{7,9} = \{T_7\}$.

In the case of the split indices, in order to accurately assign the read to this equivalence class we must:

1. determine all of the kmer alignments for a single read from each index,

2. appropriately merge overlapping kmer alignments and equivalence classes,

3. determine the set of elementary intervals[1] and the set of equivalence classes contained within those intervals,

4. and intersect all of the elementary intervals.

A single read can have multiple kmers align to multiple equivalence classes in each of the n split indices. To keep track of these alignments we store a 0-indexed interval with endpoints corresponding to kmer start positions on the read and the equivalence class corresponding to that interval. Note that the interval is closed on the left and open on the right.

After n separate alignments, we combine all of the n BUS files into one BUS file by simply remapping the equivalence class so that the set of transcripts defined by an equivalence is based on the combined transcripts from all n splits instead of just the transcripts from each separate split.

For all of the BUS records corresponding to single read, we find the set of elementary intervals and the set of transcripts corresponding to each interval, and then intersect the intervals to ultimately assign an equivalence class to the read.

The example above for the split indices would then result in the following alignment (superscript corresponds to the index number that the equivalence class is from):

1. Find split alignments. k_1: $EC_1^1 = \{T_1, T_2\}$, $EC_2^2 = \{T_7, T_9\}$ and k_2 : $EC_4^2 = \{T_5, T_6, T_7\}, EC_5^3 = \{T_{10}, T_{11})$

2. Merge. $k_1 : EC_1^1 \cup EC_2^2 = EC_{1,2} = \{T_1, T_2, T_7, T_9\}$ and $k_2 : EC_4^2 \cup EC_5^3 = EC_{4,5} = \{T_5, T_6, T_7, T_{10}, T_{11}\}$

3. Intersect. $EC = EC_{1,2} \cap EC_{7,9} = \{T_7\}$

To validate this approach, we split the human polyadenylated transcriptome, as well as an intronic sequences used for RNA velocity into two parts respectively and indexed each part. We then aligned 50 million reads to the four indices and merged the resultant BUS files as described above. Additionally we aligned the

[1] Given a list of intervals where for any interval the left endpoint is smaller than the right endpoint, an elementary interval is defined as any interval from the set of intervals constructed by taking every adjacent pair of points from a sorted list of unique endpoints. E.g. $I = \{[3, 5), [0, 4), [4, 9)\}$ and $E = \{[0, 3), [3, 4), [4, 5), [5, 9)\}$

reads to the full indices. We then computed the cell-count correlation between the quantification generated with the full indices and the quantification generated with the separate indices and found the results to be highly concordant ($r^2 = 0.97$ for counts from the polyadenylated transcriptome and $r^2 = 0.90$ for counts from the intronic sequences, Supplementary Figure 14).

The results obtained from merging BUS records that were pseudoaligned to split indices will not necessarily exactly recapitulate the results obtained from pseudoaligning to the full index. This is due to the ambiguity introduced when kmers from a single read map to multiple transcripts. When reducing the number of transcripts in the index in each split index, there are fewer sets of shared k-mers between transcripts. Given a k-mer alignment to an equivalence class in a read, the strategy in kallisto is to skip ahead in the index graph and check if the final k-mer in the read maps to the same equivalence class, a different equivalence class, or none at all. This skip ahead strategy, while appropriate for the full index, can skip intermediate k-mer alignments that only result from an equivalence class in the full transcriptome, thereby resulting in fewer alignments and a slight loss in pseudoalignments when splitting indices and subsequently merging results (Supplementary Figure 14).

References

[GKO14] Dominic Grün, Lennart Kester, and Alexander van Oudenaarden. "Validation of noise models for single-cell transcriptomics." In: *Nature Methods* 11.6 (June 2014), pp. 637–640. DOI: `10.1038/nmeth.2930`.

[Bra+16] Nicolas L Bray et al. "Near-optimal probabilistic RNA-seq quantification." In: *Nature Biotechnology* 34.5 (Apr. 2016), pp. 525–527. ISSN: 1087-0156. DOI: `10.1038/nbt.3519`.

[MNP19] Páll Melsted, Vasilis Ntranos, and Lior Pachter. "The Barcode, UMI, Set format and BUStools." In: *Bioinformatics* (May 2019). ISSN: 1367-4803. DOI: `10.1093/bioinformatics/btz279`.