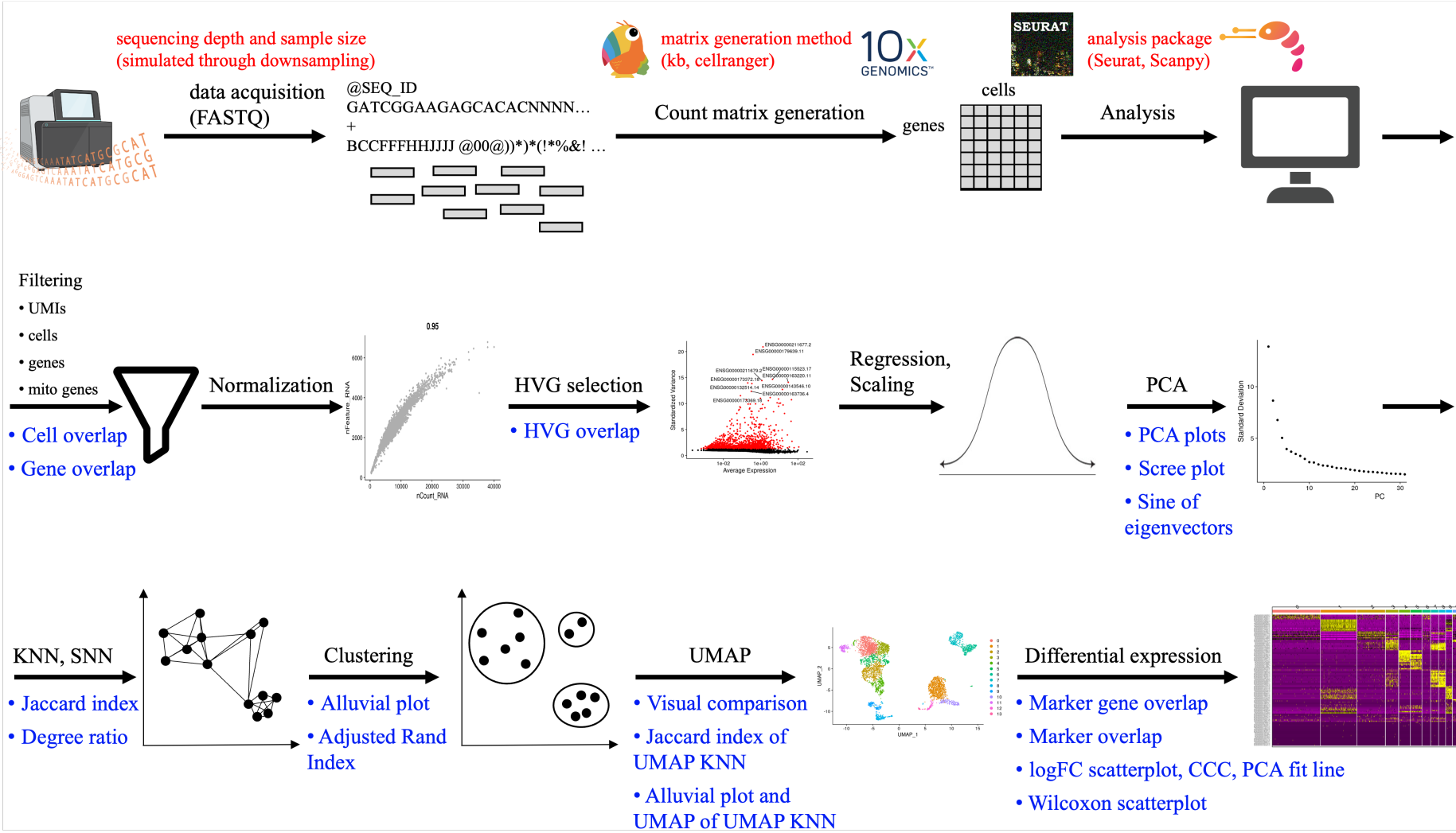
Supplemental Information
The impact of package selection and versioning on single-cell RNA-seq analysis

Joseph M Rich[1,2], Lambda Moses[1], Pétur Helgi Einarsson[3], Kayla Jackson[1,2], Laura Luebbert[1], A. Sina Booeshaghi[1], Sindri Antonsson[3], Delaney K. Sullivan[1,4], Nicolas Bray[5], Páll Melsted[3], and Lior Pachter[*1,6,7]

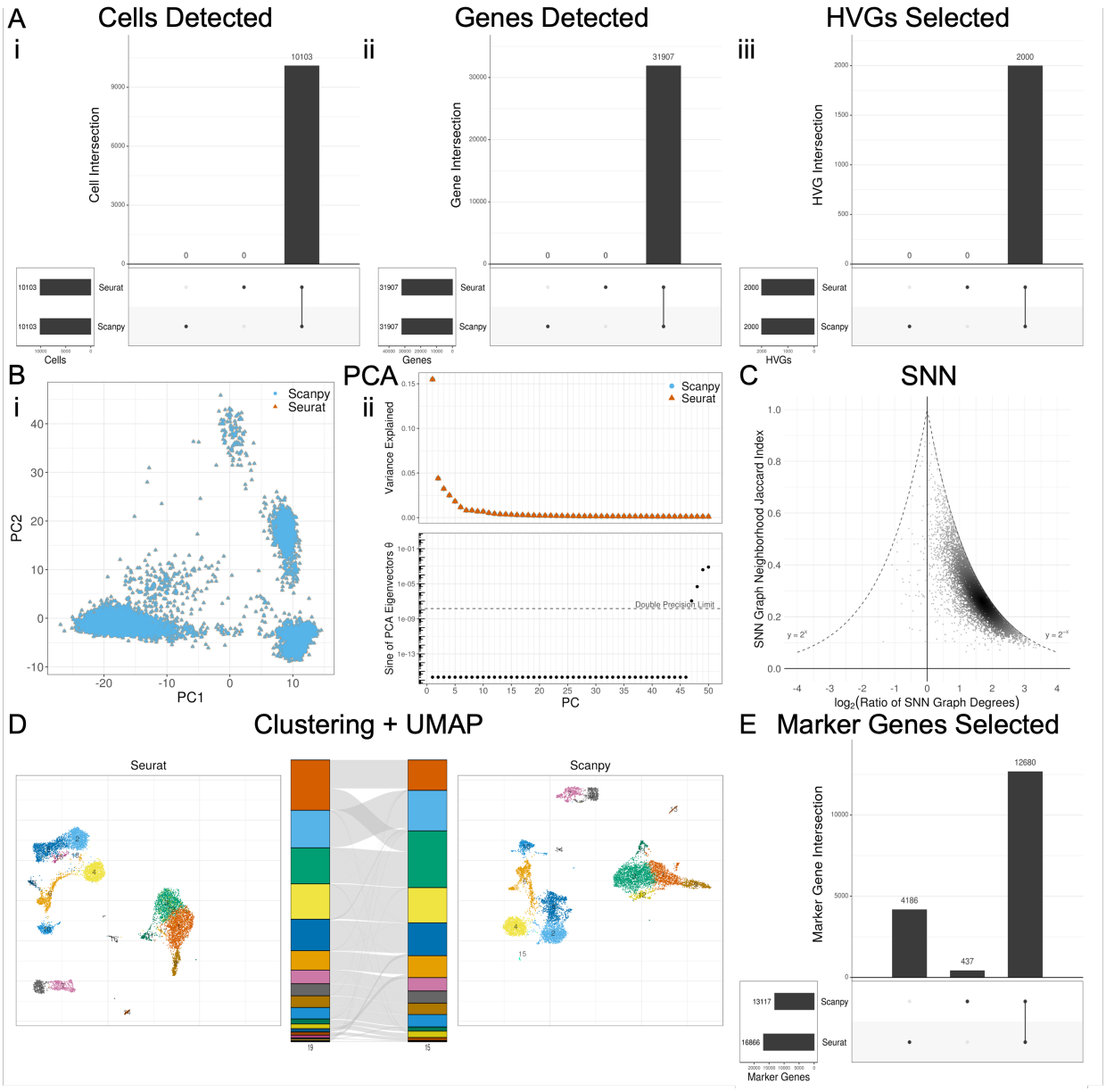
[1]*Biology and Biological Engineering, California Institute of Technology, Pasadena, CA, 91125, USA*
[2]*USC-Caltech MD/PhD Program, Keck School of Medicine, Los Angeles, CA, 90033, USA*
[3]*Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, Reykjavík, Iceland*
[4]*UCLA-Caltech Medical Scientist Training Program, David Geffen School of Medicine, University of California, Los Angeles, Los Angeles, CA, 90095, USA*
[5]*Boston, MA*
[6]*Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, 91125, USA*
[7]*Lead Contact*

April 9, 2024

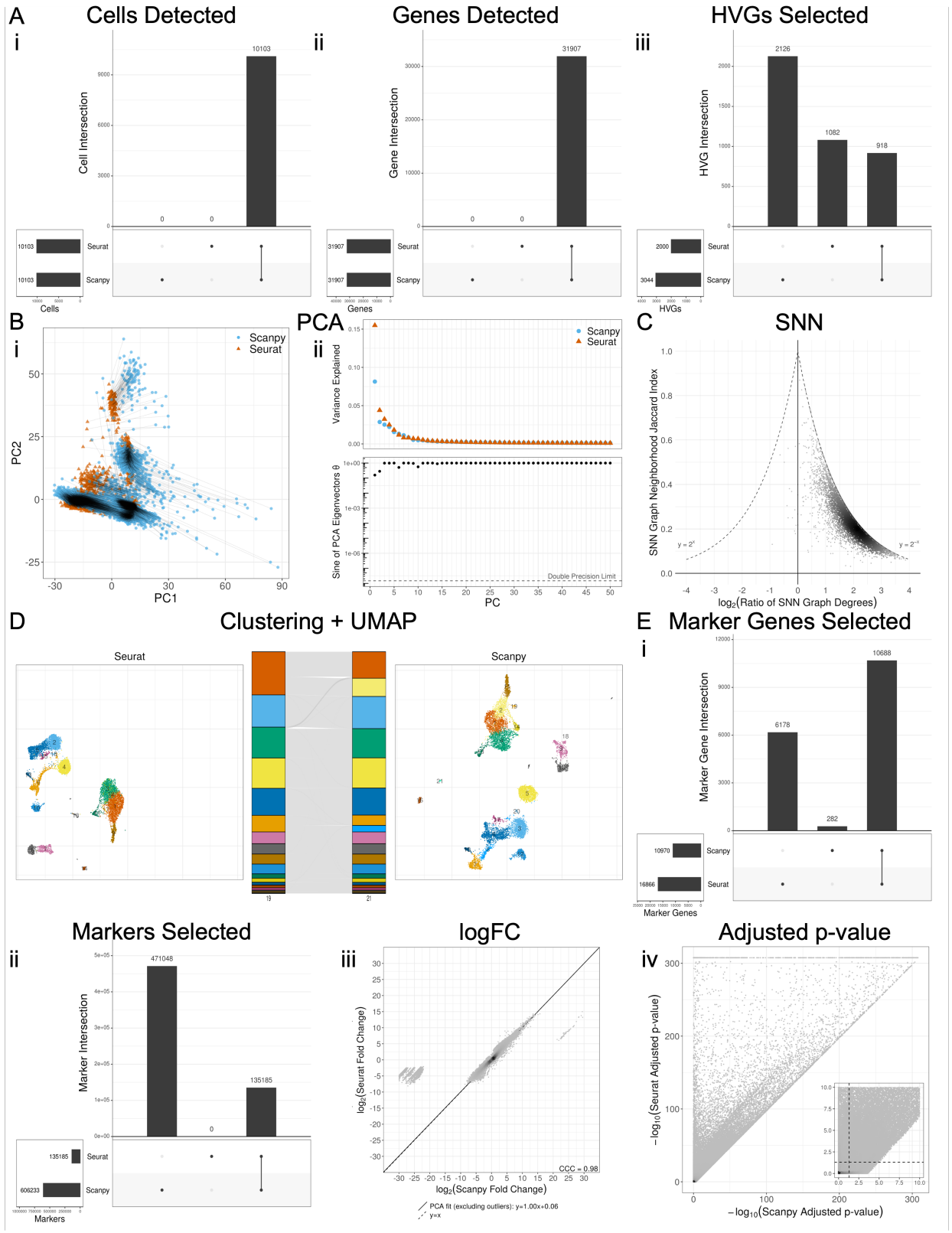
[*]Correspondence: lpachter@caltech.edu.


sequencing depth and sample size
(simulated through downsampling)
matrix generation method
(kb, cellranger)
10x GENOMICS
SEURAT
analysis package
(Seurat, Scanpy)
data acquisition
(FASTQ)
@SEQ_ID
GATCGGAAGAGCACACNNNN…
+
BCCFFFHHJJJJ @00@))*)*(!*%&! …
Count matrix generation
cells
genes
Analysis
Filtering
• UMIs
• cells
• genes
• mito genes
• Cell overlap
• Gene overlap
Normalization
0.95
nFeature_RNA
nCount_RNA
HVG selection
• HVG overlap
Standardized Variance
Average Expression
Regression,
Scaling
PCA
• PCA plots
• Scree plot
• Sine of
eigenvectors
Standard Deviation
PC
KNN, SNN
• Jaccard index
• Degree ratio
Clustering
• Alluvial plot
• Adjusted Rand
Index
UMAP
• Visual comparison
• Jaccard index of
UMAP KNN
• Alluvial plot and
UMAP of UMAP KNN
UMAP_1
UMAP_2
Differential expression
• Marker gene overlap
• Marker overlap
• logFC scatterplot, CCC, PCA fit line
• Wilcoxon scatterplot

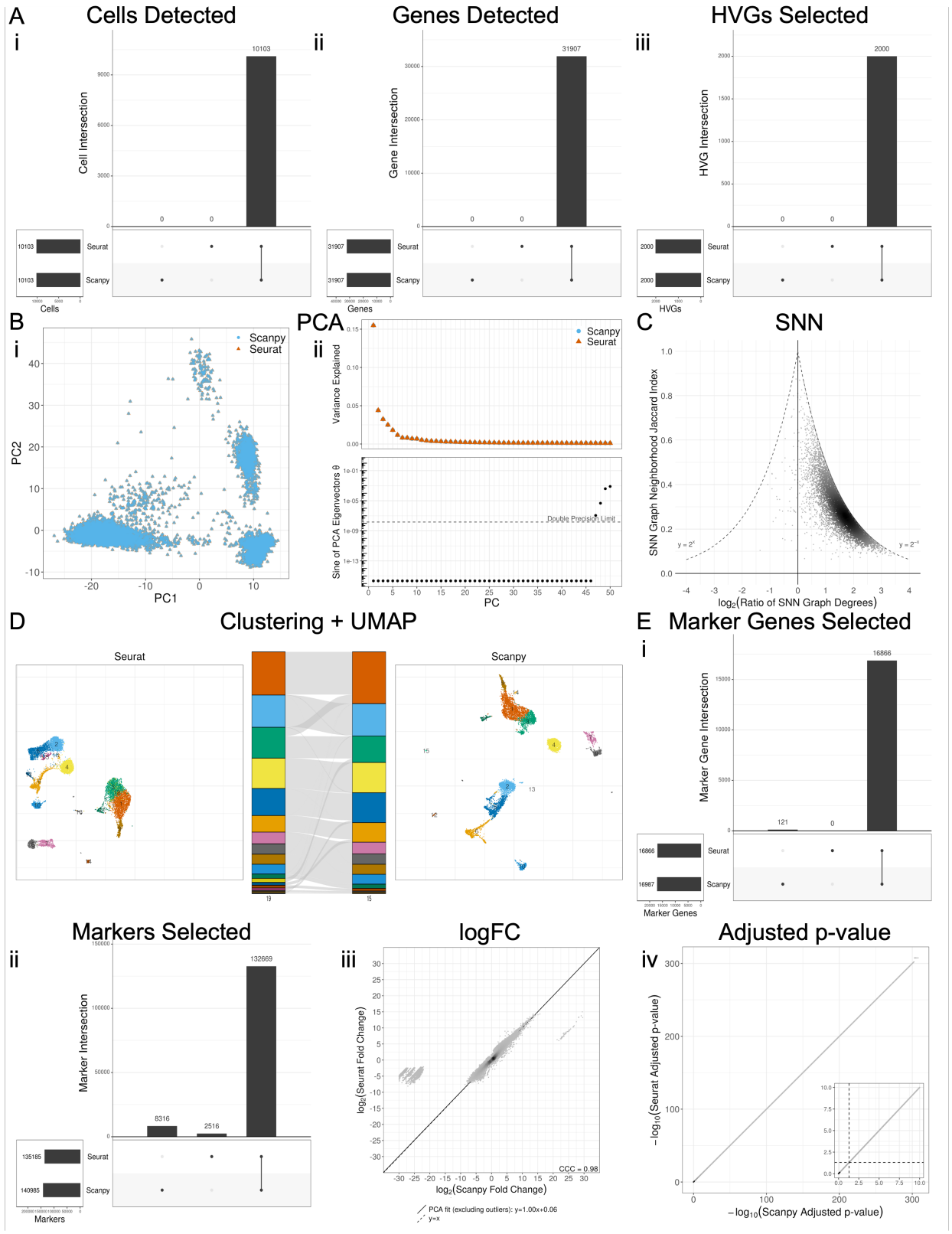
Supplemental Figure 1: scRNA-seq matrix generation and analysis overview. Red = sources of variability explored in this study; Blue = plots and metrics used to assess function similarity



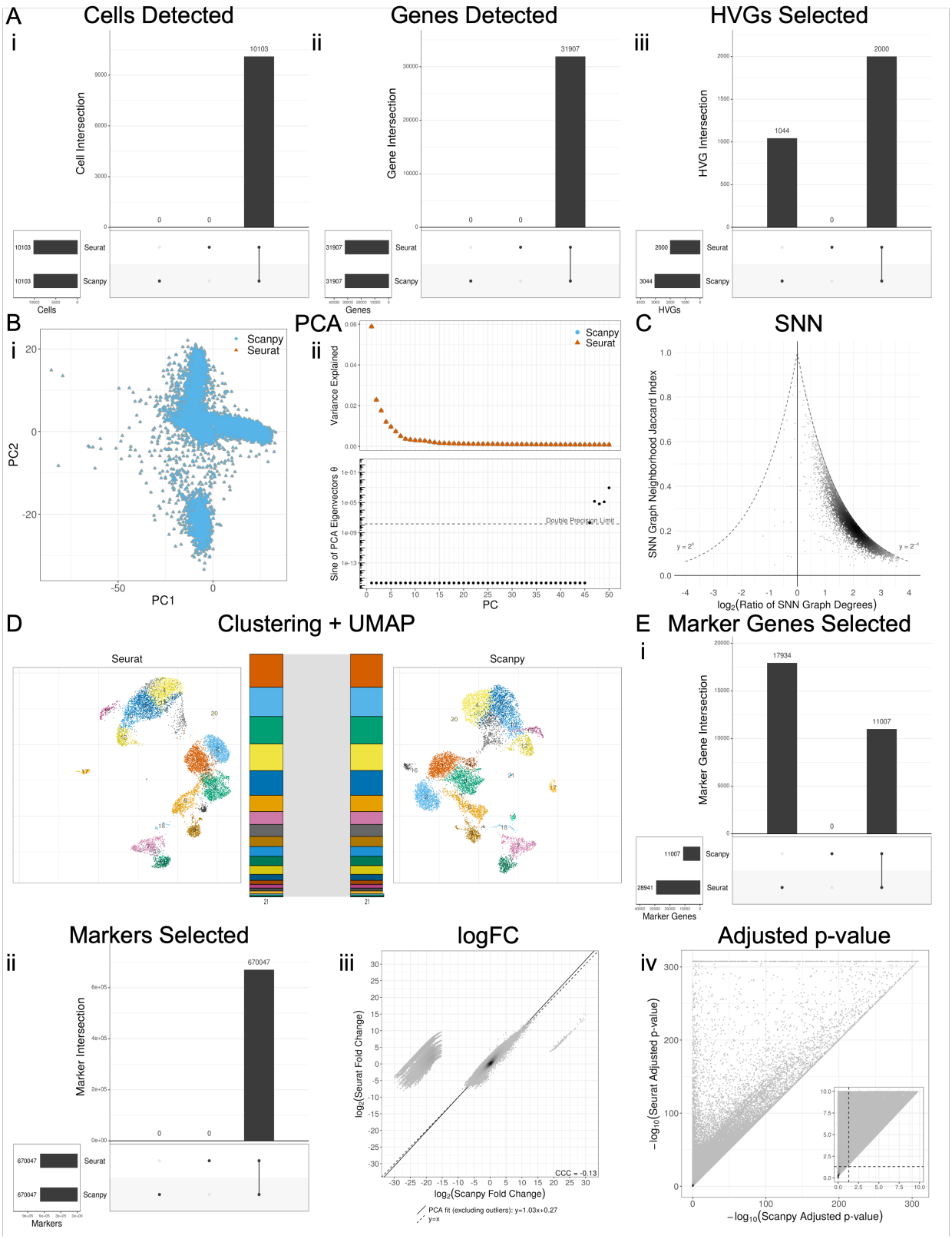
Supplemental Figure 2: Seurat vs. Scanpy, aligned function arguments (Seurat-like), sequential analysis. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (Seurat/Scanpy) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis UpSet plot through overlap of all significant ($p < 0.05$) marker genes across all clusters.


A
Cells Detected
Genes Detected
HVGs Selected
i
ii
iii
Cell Intersection
Gene Intersection
HVG Intersection
10103
31907
2126
1082
918
Seurat
Scanpy
Cells
Genes
HVGs
B
PCA
C
SNN
Scanpy
Seurat
PC1
PC2
Variance Explained
Sine of PCA Eigenvectors θ
Double Precision Limit
PC
SNN Graph Neighborhood Jaccard Index
log2(Ratio of SNN Graph Degrees)
D
Clustering + UMAP
Seurat
Scanpy
E
Marker Genes Selected
Marker Gene Intersection
10688
6178
282
Marker Genes
Markers Selected
Marker Intersection
471048
135185
Markers
logFC
log2(Seurat Fold Change)
log2(Scanpy Fold Change)
CCC = 0.98
PCA fit (excluding outliers): y=1.00x+0.06
y=x
Adjusted p-value
−log10(Seurat Adjusted p-value)
−log10(Scanpy Adjusted p-value)

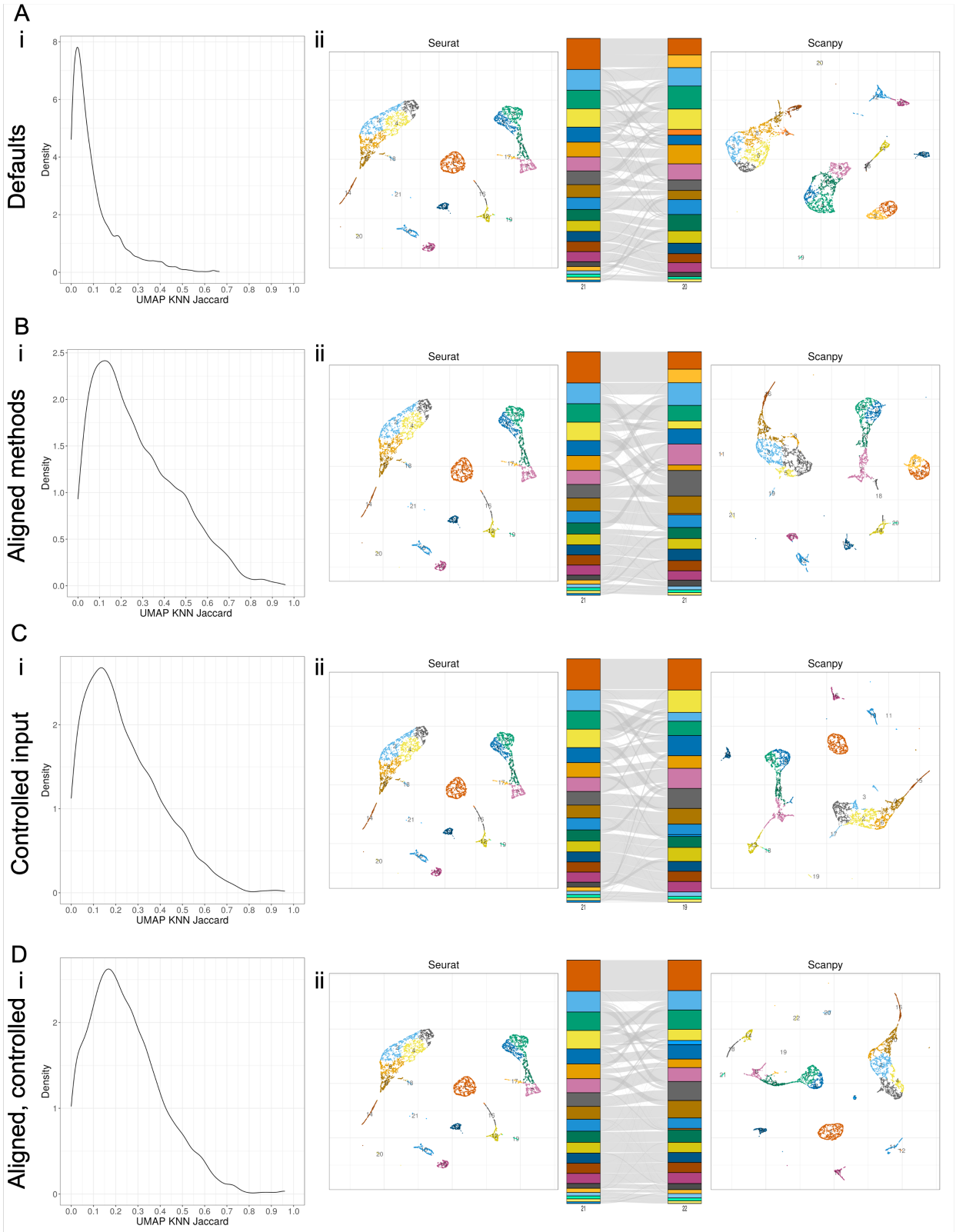
Supplemental Figure 3: Seurat vs. Scanpy, default function arguments, controlled analysis. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (Seurat/Scanpy) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis through overlap of all significant ($p < 0.05$) marker genes across all clusters. (F) Differential expression analysis with aligned clusters of overlap of all markers. (G) Differential expression analysis with aligned clusters of logFC similarity. (H) Differential expression analysis with aligned clusters of adjusted p-value similarity.


A
Cells Detected
Genes Detected
HVGs Selected
i
ii
iii
Cell Intersection
Gene Intersection
HVG Intersection
10103
31907
2000
Seurat
Scanpy
Cells
Genes
HVGs
B
PCA
C
SNN
i
ii
PC2
PC1
Variance Explained
Sine of PCA Eigenvectors θ
Double Precision Limit
PC
SNN Graph Neighborhood Jaccard Index
log2(Ratio of SNN Graph Degrees)
D
Clustering + UMAP
Seurat
Scanpy
E
Marker Genes Selected
Marker Gene Intersection
16866
121
16987
Marker Genes
Markers Selected
ii
Marker Intersection
132669
8316
2516
135185
140985
Markers
logFC
iii
log2(Seurat Fold Change)
log2(Scanpy Fold Change)
CCC = 0.98
PCA fit (excluding outliers): y=1.00x+0.06
y=x
Adjusted p-value
iv
−log10(Seurat Adjusted p-value)
−log10(Scanpy Adjusted p-value)

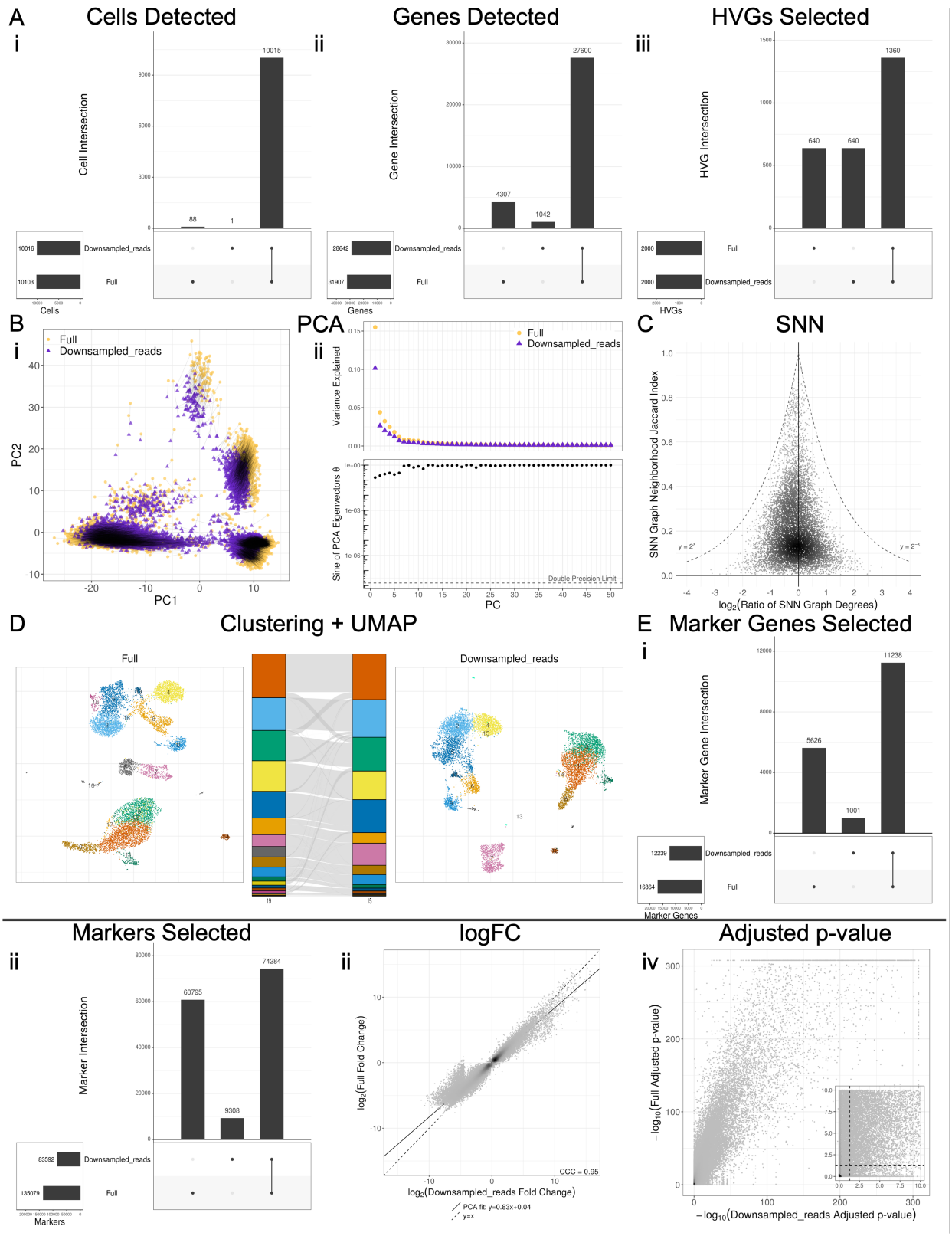
Supplemental Figure 4: Seurat vs. Scanpy, aligned function arguments (Seurat-like), controlled analysis. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (Seurat/Scanpy) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis through overlap of all significant ($p < 0.05$) marker genes across all clusters. (F) Differential expression analysis with aligned clusters of overlap of all markers. (G) Differential expression analysis with aligned clusters of logFC similarity. (H) Differential expression analysis with aligned clusters of adjusted p-value similarity.


A
Cells Detected
Genes Detected
HVGs Selected
i
ii
iii
Cell Intersection
Gene Intersection
HVG Intersection
10103
31907
2000
1044
Seurat
Scanpy
Cells
Genes
HVGs
B
PCA
C
SNN
PC1
PC2
Variance Explained
Sine of PCA Eigenvectors θ
Double Precision Limit
PC
SNN Graph Neighborhood Jaccard Index
log2(Ratio of SNN Graph Degrees)
D
Clustering + UMAP
E
Marker Genes Selected
Marker Gene Intersection
17934
11007
28941
Marker Genes
Markers Selected
Marker Intersection
670047
Markers
logFC
log2(Seurat Fold Change)
log2(Scanpy Fold Change)
CCC = -0.13
PCA fit (excluding outliers): y=1.03x+0.27
y=x
Adjusted p-value
−log10(Seurat Adjusted p-value)
−log10(Scanpy Adjusted p-value)

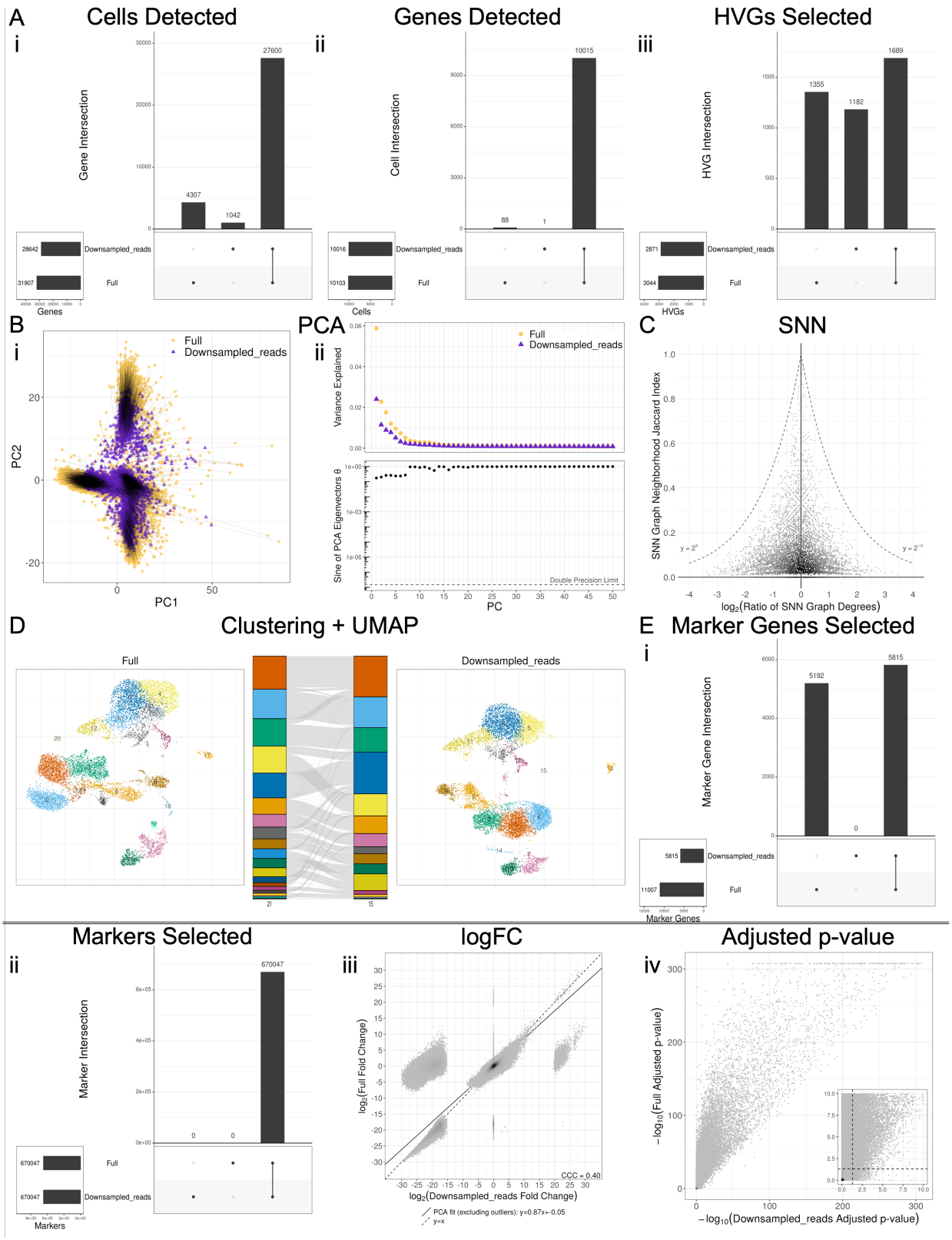
Supplemental Figure 5: Seurat vs. Scanpy, aligned function arguments (Scanpy-like), controlled analysis. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (Seurat/Scanpy) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis through overlap of all significant ($p < 0.05$) marker genes across all clusters. (F) Differential expression analysis with aligned clusters of overlap of all markers. (G) Differential expression analysis with aligned clusters of logFC similarity. (H) Differential expression analysis with aligned clusters of adjusted p-value similarity.


A
Defaults
i
ii
Seurat
Scanpy
Density
UMAP KNN Jaccard
B
Aligned methods
C
Controlled input
D
Aligned, controlled

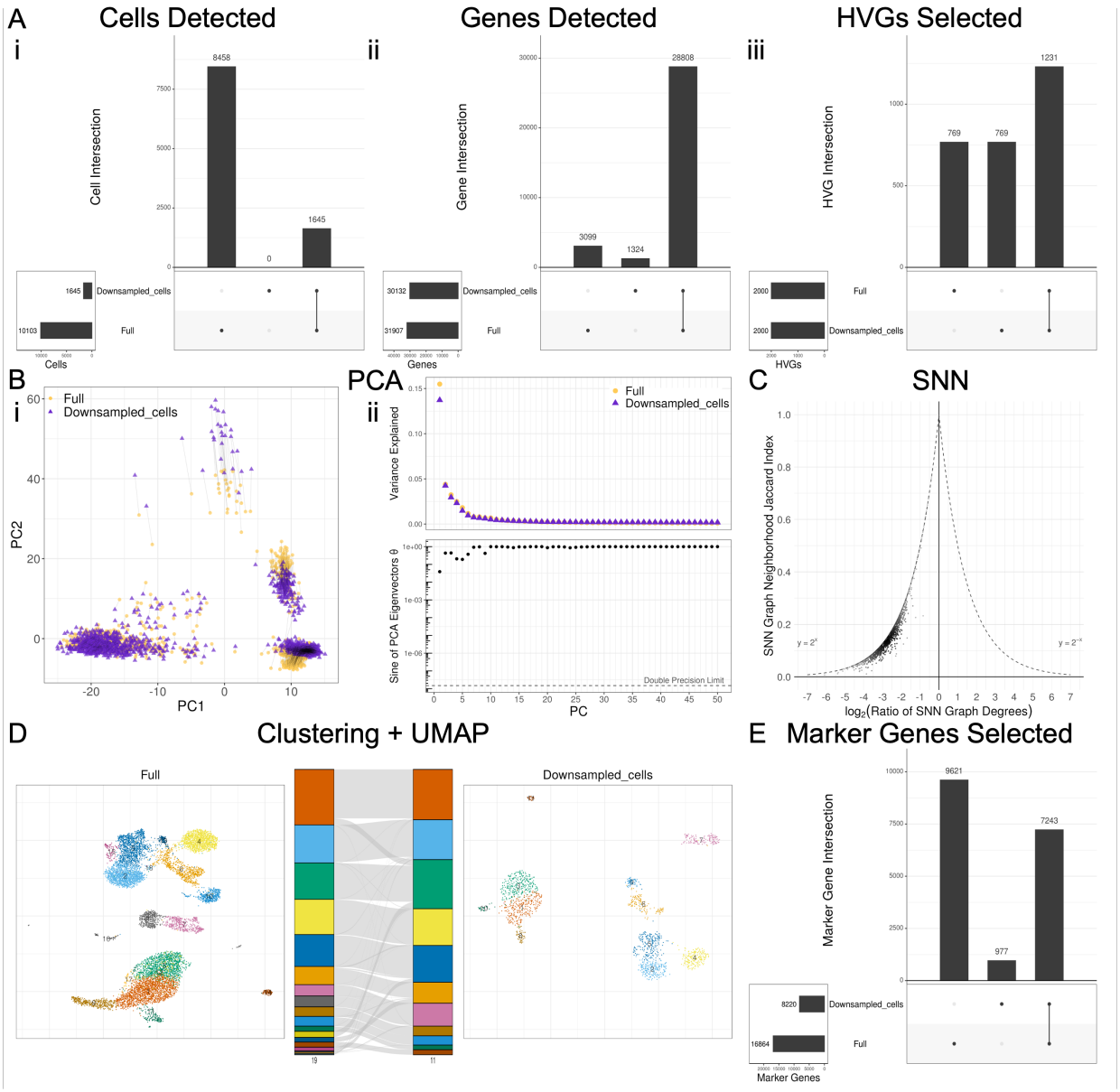
Supplemental Figure 6: Extended UMAP analysis for Seurat vs. Scanpy showing density plot of UMAP-derived KNN graph neighborhood Jaccard indices, alluvial plot of leiden clustering performed on UMAP-derived KNN graph, and UMAP projection of UMAP-derived KNN graph colored with leiden clustering results. (A) Default methods, sequential analysis. (B) Aligned methods, sequential analysis. (C) Default methods, controlled analysis. (D) Aligned methods, controlled analysis.


A
Cells Detected
Genes Detected
HVGs Selected
i
ii
iii
Cell Intersection
Gene Intersection
HVG Intersection
10015
88
1
27600
4307
1042
1360
640
640
10016
10103
28642
31907
2000
2000
Downsampled_reads
Full
Cells
Genes
HVGs
B
PCA
C
SNN
Full
Downsampled_reads
PC2
PC1
Variance Explained
Sine of PCA Eigenvectors θ
Double Precision Limit
PC
SNN Graph Neighborhood Jaccard Index
log2(Ratio of SNN Graph Degrees)
D
Clustering + UMAP
E
Marker Genes Selected
Marker Gene Intersection
11238
5626
1001
12239
16864
Marker Genes
Markers Selected
Marker Intersection
74284
60795
9308
83592
135079
Markers
logFC
log2(Full Fold Change)
log2(Downsampled_reads Fold Change)
CCC = 0.95
PCA fit: y=0.83x+0.04
y=x
Adjusted p-value
−log10(Full Adjusted p-value)
−log10(Downsampled_reads Adjusted p-value)

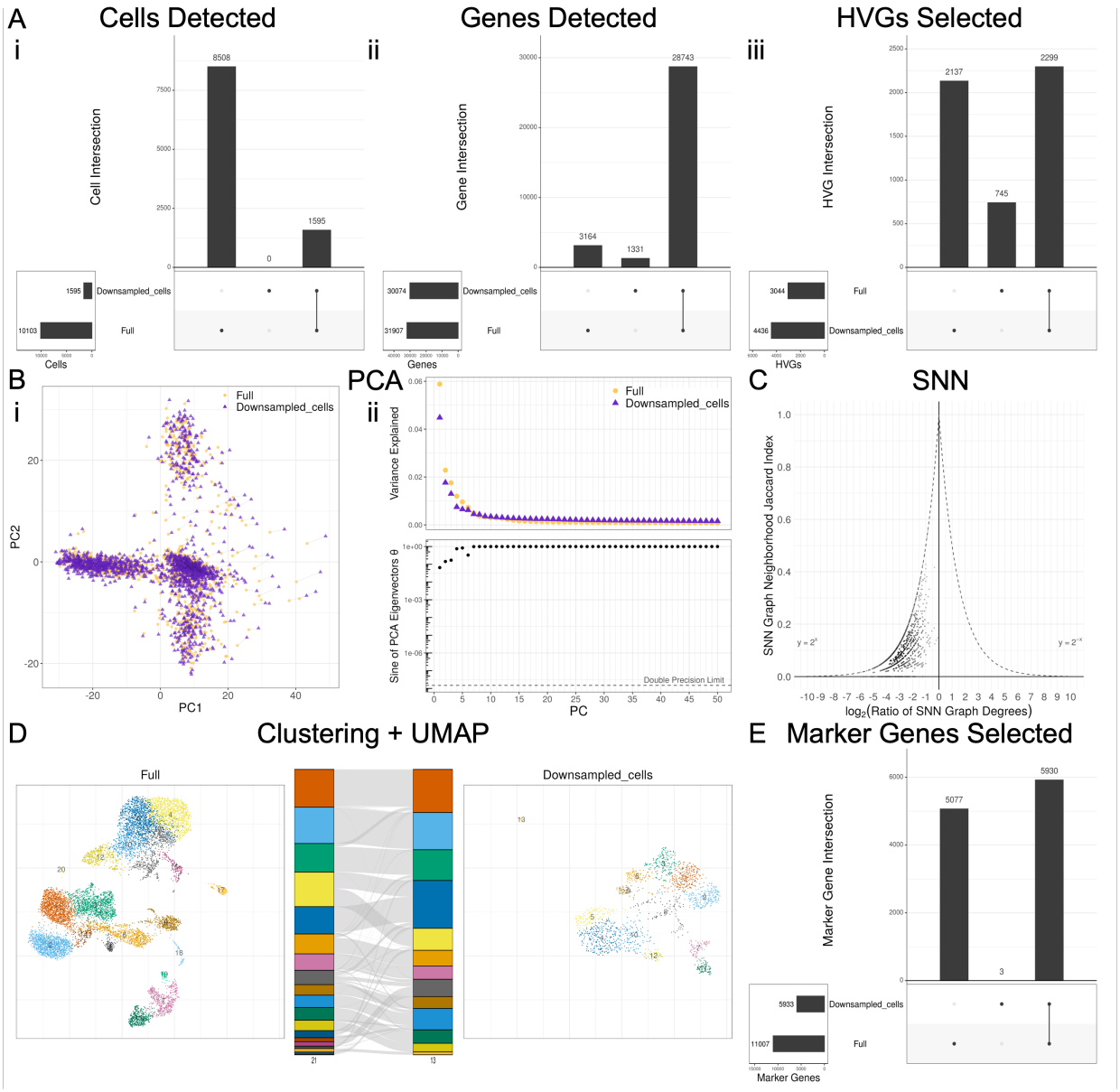
Supplemental Figure 7: Read downsampling to 4% of original size vs. full dataset in Seurat, with a fraction that displays variation comparable to Seurat vs. Scanpy. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (full/downsampled) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis through overlap of all significant ($p < 0.05$) marker genes across all clusters. (F) Differential expression analysis with aligned clusters of overlap of all markers. (G) Differential expression analysis with aligned clusters of logFC similarity. (H) Differential expression analysis with aligned clusters of adjusted p-value similarity.


A
Cells Detected
Genes Detected
HVGs Selected
i
ii
iii
Gene Intersection
Cell Intersection
HVG Intersection
27600
4307
1042
10015
88
1
1689
1355
1182
28642
31907
Downsampled_reads
Full
Genes
10016
10103
Cells
2871
3044
HVGs
B
PCA
C
SNN
Full
Downsampled_reads
PC1
PC2
Variance Explained
Sine of PCA Eigenvectors θ
Double Precision Limit
PC
SNN Graph Neighborhood Jaccard Index
log2(Ratio of SNN Graph Degrees)
D
Clustering + UMAP
E
Marker Genes Selected
Marker Gene Intersection
5192
5815
0
11007
Marker Genes
Markers Selected
Marker Intersection
670047
0
Markers
logFC
Adjusted p-value
log2(Full Fold Change)
log2(Downsampled_reads Fold Change)
CCC = 0.40
PCA fit (excluding outliers): y=0.87x+-0.05
y=x
−log10(Full Adjusted p-value)
−log10(Downsampled_reads Adjusted p-value)

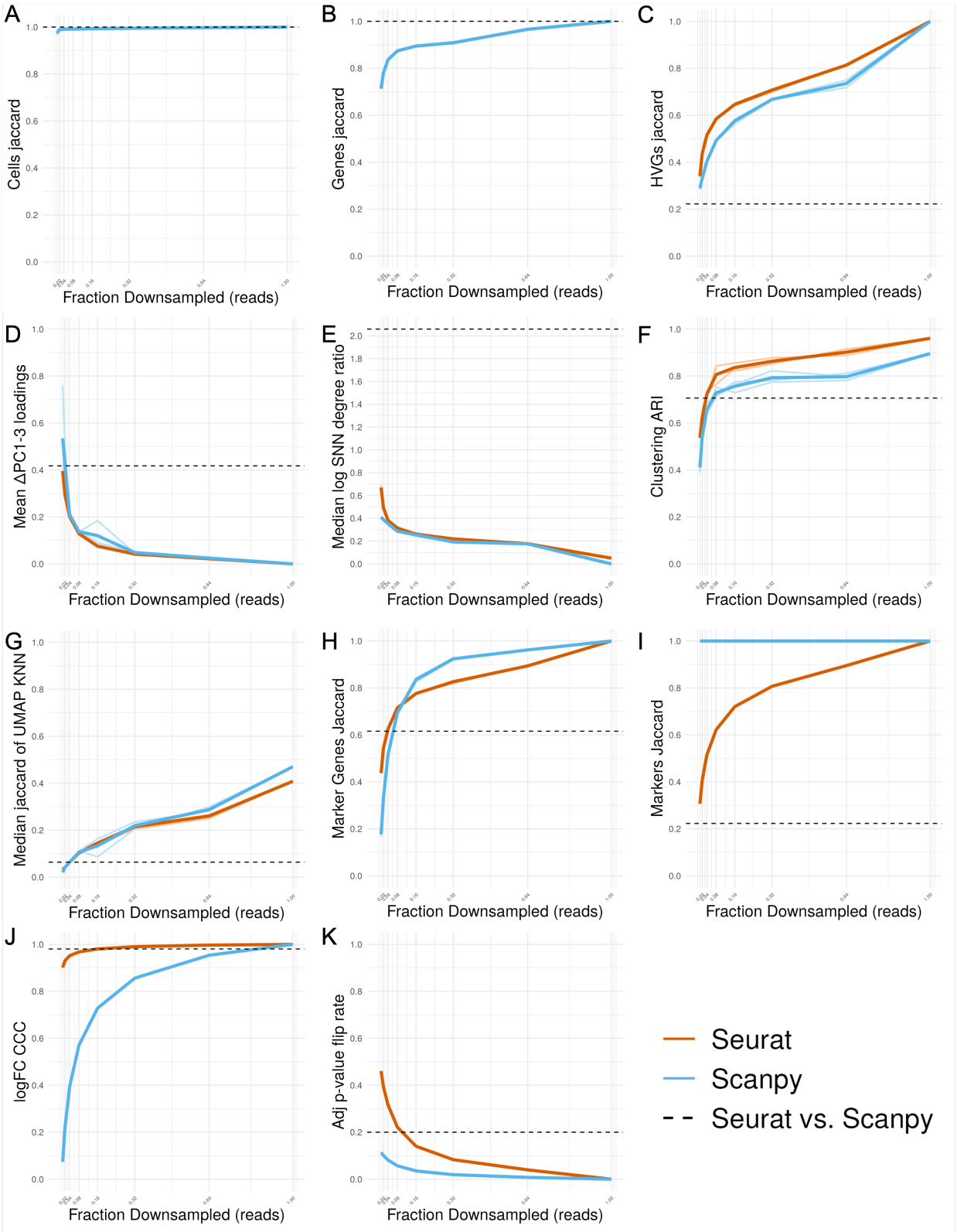
Supplemental Figure 8: Read downsampling to 4% of original size vs. full dataset in Scanpy, with a fraction that displays variation comparable to Seurat vs. Scanpy. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (full/-downsampled) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis through overlap of all significant ($p < 0.05$) marker genes across all clusters. (F) Differential expression analysis with aligned clusters of overlap of all markers. (G) Differential expression analysis with aligned clusters of logFC similarity. (H) Differential expression analysis with aligned clusters of adjusted p-value similarity.



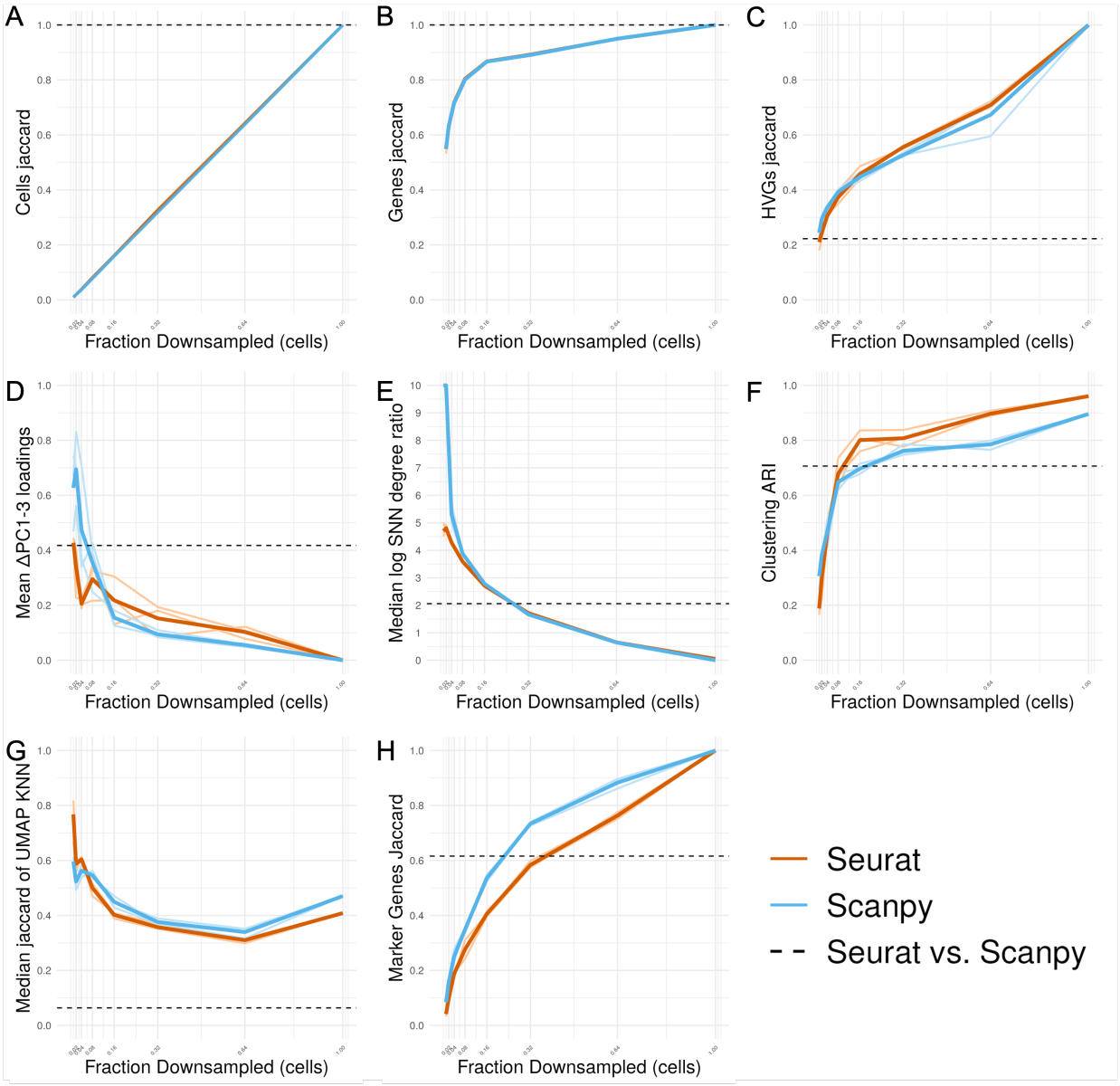
Supplemental Figure 9: Cell downsampling to 16% of original size vs. full dataset in Seurat, with a fraction that displays variation comparable to Seurat vs. Scanpy. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (full/downsampled) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis UpSet plot through overlap of all significant ($p < 0.05$) marker genes across all clusters.



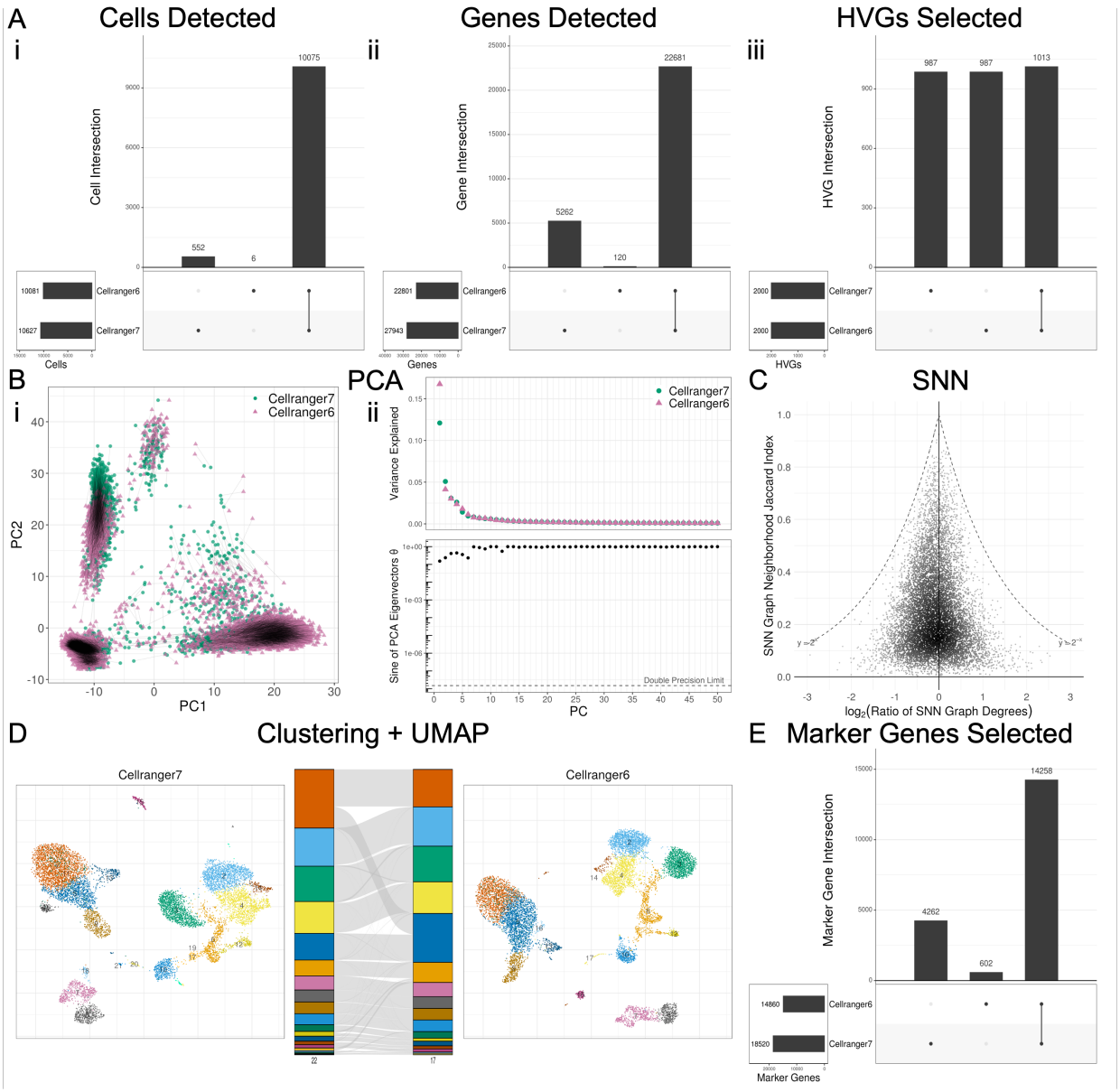
Supplemental Figure 10: Cell downsampling to 16% of original size vs. full dataset in Scanpy, with a fraction that displays variation comparable to Seurat vs. Scanpy. (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. Black lines = point mapping between conditions. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (full/-downsampled) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis UpSet plot through overlap of all significant ($p < 0.05$) marker genes across all clusters.


A
Cells jaccard
Fraction Downsampled (reads)
B
Genes jaccard
Fraction Downsampled (reads)
C
HVGs jaccard
Fraction Downsampled (reads)
D
Mean ΔPC1-3 loadings
Fraction Downsampled (reads)
E
Median log SNN degree ratio
Fraction Downsampled (reads)
F
Clustering ARI
Fraction Downsampled (reads)
G
Median jaccard of UMAP KNN
Fraction Downsampled (reads)
H
Marker Genes Jaccard
Fraction Downsampled (reads)
I
Markers Jaccard
Fraction Downsampled (reads)
J
logFC CCC
Fraction Downsampled (reads)
K
Adj p-value flip rate
Fraction Downsampled (reads)
Seurat
Scanpy
Seurat vs. Scanpy

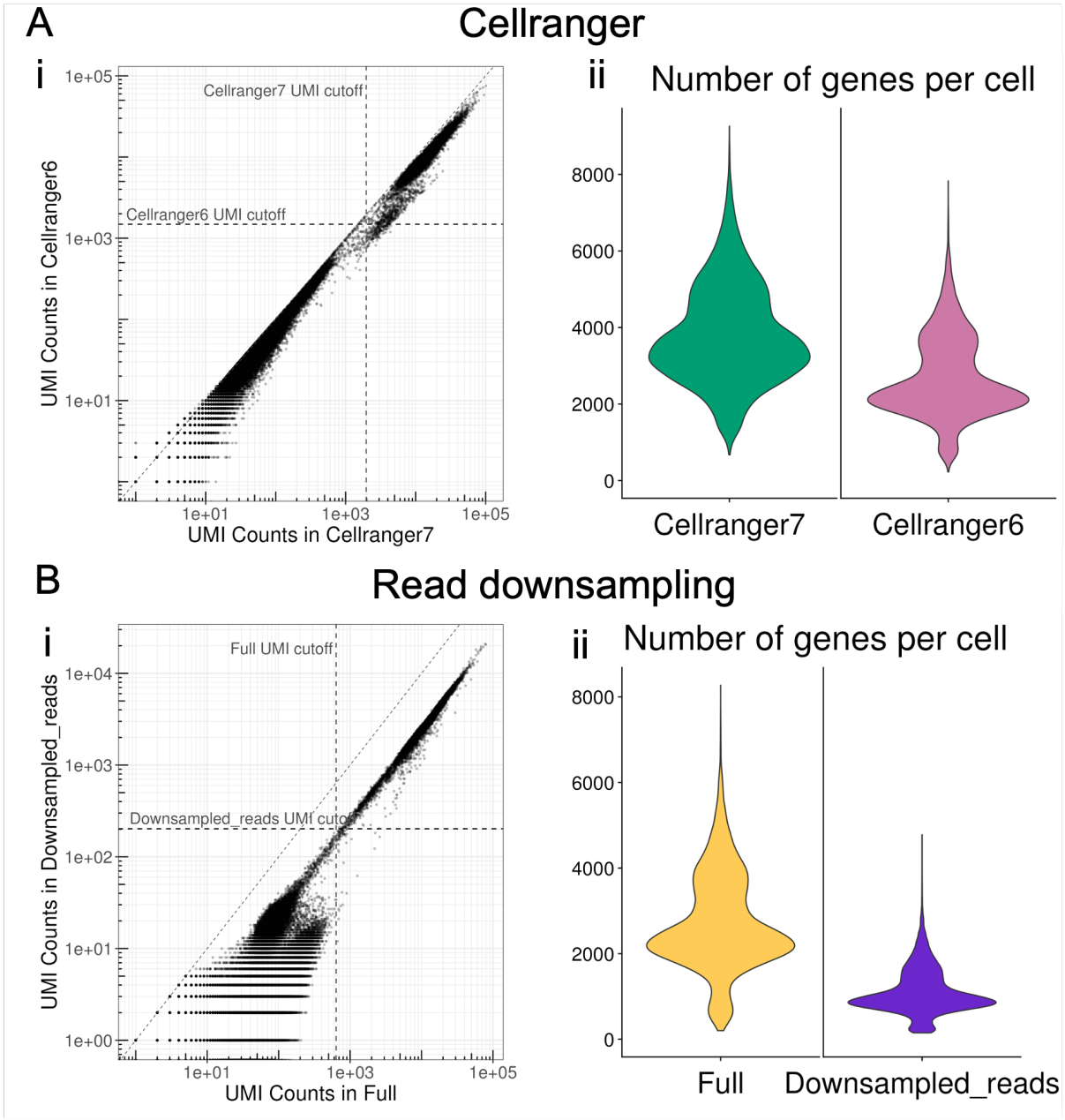
Supplemental Figure 11: Metrics vs. fraction of reads downsampled. (A) Jaccard index of cell overlap. (B) Jaccard index of gene overlap. (C) Jaccard index of HVG overlap. (D) Mean of difference in corresponding PC loadings for PCs 1-3. (E) Median log(SNN degree ratio). (F) Adjusted Rand Index of clusters. (G) Median Jaccard index of UMAP-derived KNN graphs for each cell. (H) Jaccard index of sets of significant ($p < 0.05$) marker genes across all clusters. (I) Jaccard index of all markers. (J) CCC of logFC scatterploz. (K) Fraction of adjusted p-value that flipped across p=0.05 threshold. Orange = Seurat; Blue = Scanpy; lighter lines = replicates of downsampling across random seeds; dashed line = value recorded by Seurat vs. Scanpy with default settings.



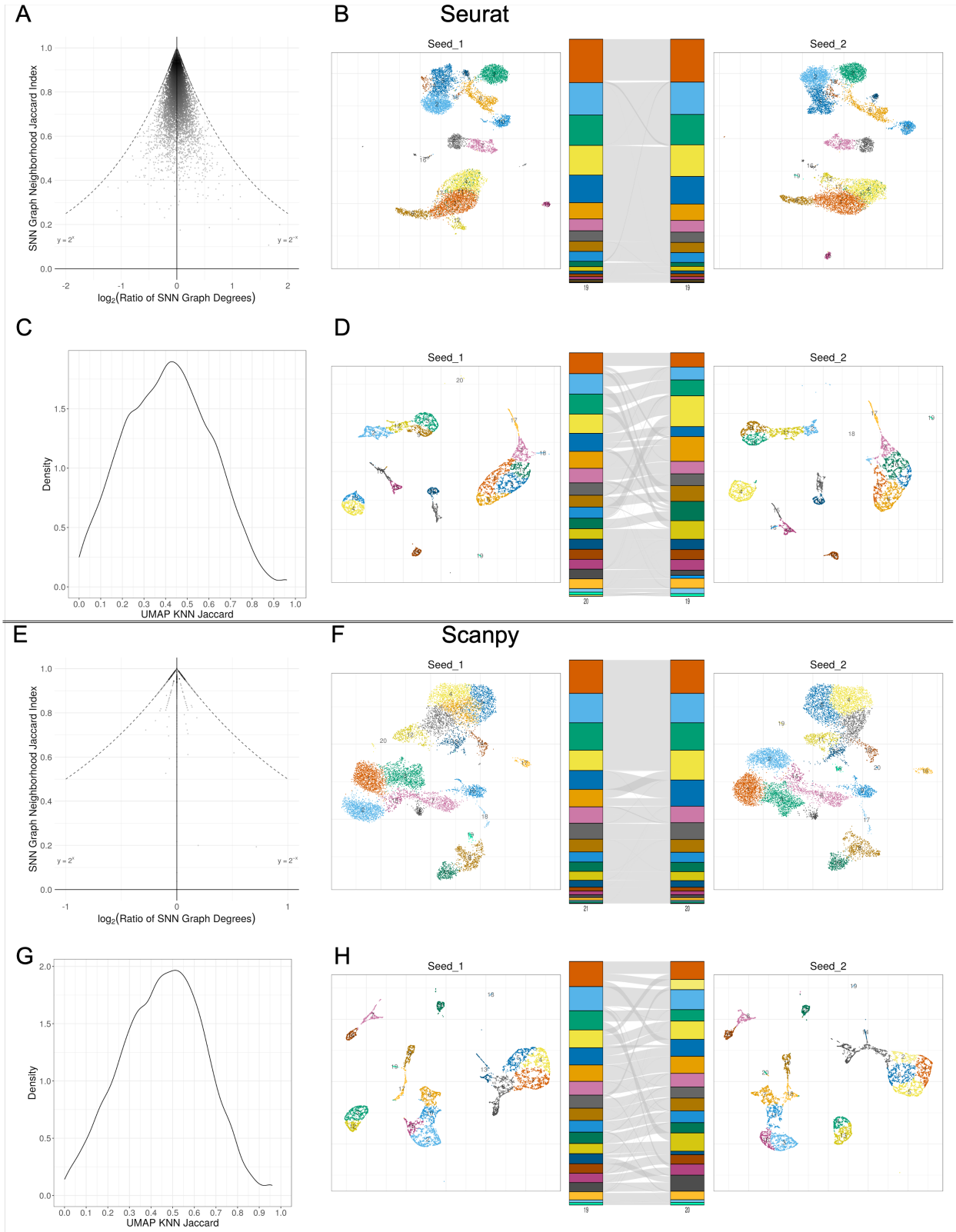
Supplemental Figure 12: Metrics vs. fraction of cells downsampled. (A) Jaccard index of cell overlap. (B) Jaccard index of gene overlap. (C) Jaccard index of HVG overlap. (D) Mean of difference in corresponding PC loadings for PCs 1-3. (E) Median log(SNN degree ratio). (F) Adjusted Rand Index of clusters. (G) Median Jaccard index of UMAP-derived KNN graphs for each cell. (H) Jaccard index of sets of significant ($p < 0.05$) marker genes across all clusters. (I) Jaccard index of all markers. (J) CCC of logFC scatterplot. (K) Fraction of adjusted p-value that flipped across p=0.05 threshold. Orange = Seurat; Blue = Scanpy; lighter lines = replicates of downsampling across random seeds; dashed line = value recorded by Seurat vs. Scanpy with default settings.



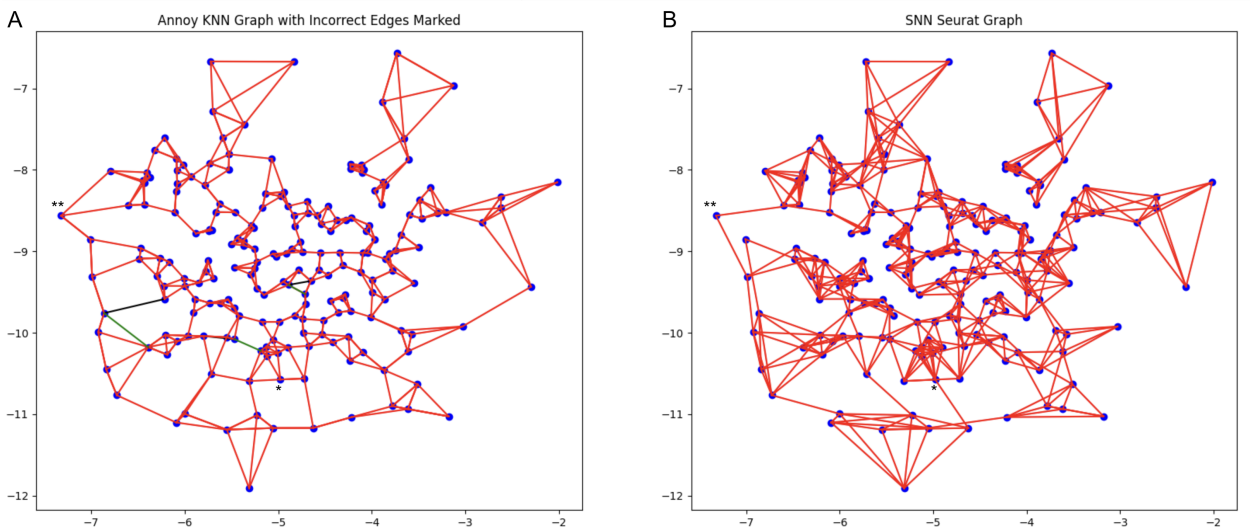
Supplemental Figure 13: Cell Ranger version control (v7 vs. v6). (A) Filtering and HVG selection analysis UpSet plots consisting of overlap of sets of cells, genes, and HVGs. (B) PCA analysis through projection onto first 2 PCs, Scree plot eigenvalue comparison, and sine of eigenvectors. (C) KNN/SNN analysis through SNN neighborhood Jaccard index and degree ratio (v7/v6) per cell. (D) Clustering and UMAP analysis through UMAP plots of each condition, with alluvial plot showing cluster assignment mapping and degree of agreement. (E) Differential expression analysis through overlap of all significant ($p < 0.05$) marker genes across all clusters.



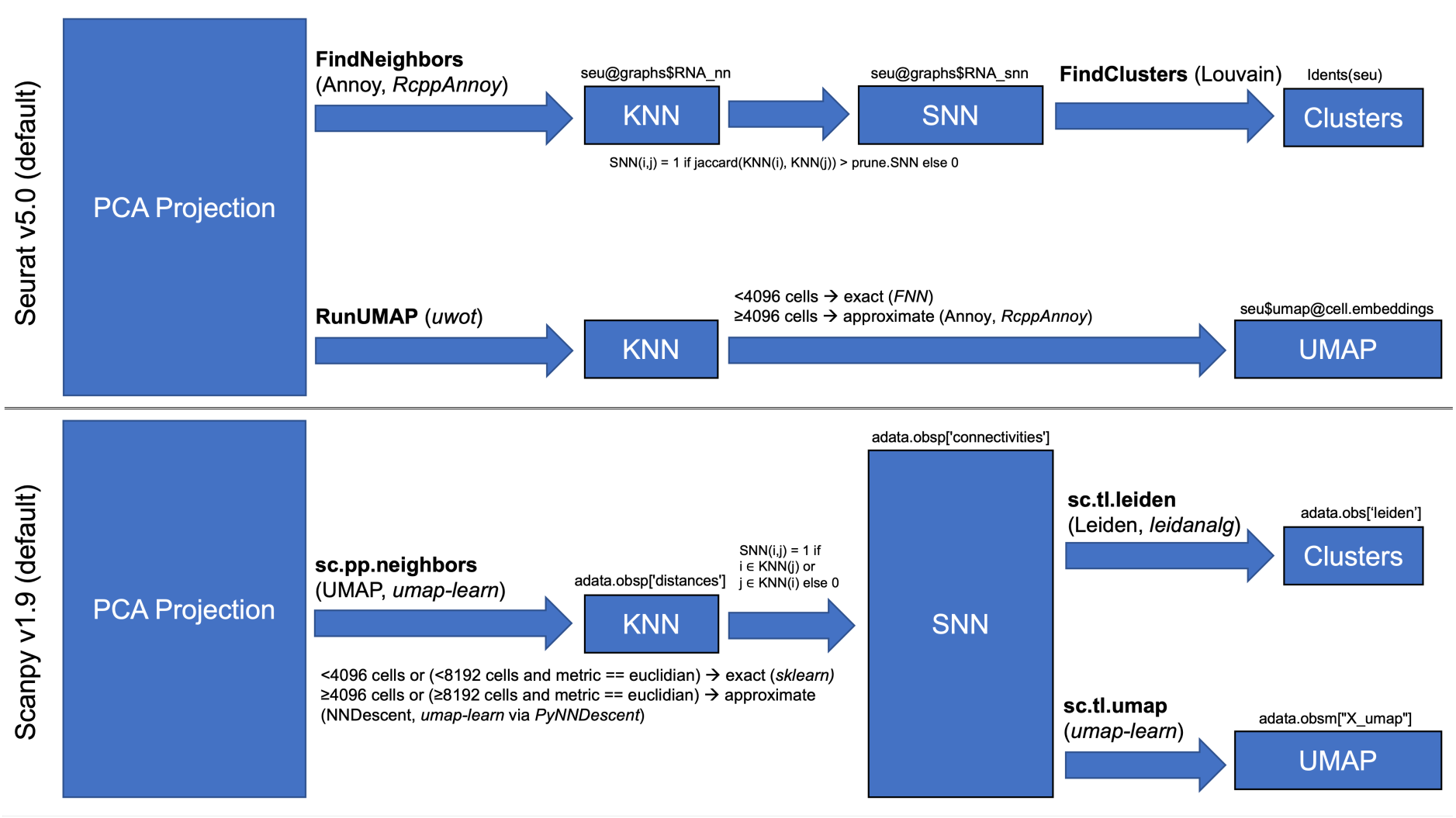
Supplemental Figure 14: UMI counts pre-filtering and number of features under different conditions. (A) Cell Ranger v7 vs. v6. (B) Full size dataset vs. downsampled reads to 4%.


A
B
Seurat
SNN Graph Neighborhood Jaccard Index
log2(Ratio of SNN Graph Degrees)
y = 2^x
y = 2^-x
Seed_1
Seed_2
C
D
Density
UMAP KNN Jaccard
E
F
Scanpy
G
H

Supplemental Figure 15: Variability introduced by differences in random seeds for functions involving randomness. (A) Seurat SNN graph generation (Annoy), with different seeds for KNN graph generation. (B) Seurat clustering (Louvain), with different seeds for Louvain; and UMAP-projection (uwot), with different seeds for UMAP projection. (C) Seurat KNN graph generation from UMAP space, with different seeds for UMAP projection. (D) Seurat clustering (Leiden) and UMAP projection from UMAP space, with different seeds for UMAP projection. (E) Scanpy SNN graph generation (NNDescent), with different seeds for KNN graph generation. (F) Scanpy clustering (Leiden), with different seeds for Leiden; and UMAP-projection (UMAP-learn), with different seeds for UMAP projection. (G) Scanpy KNN graph generation from UMAP space, with different seeds for UMAP projection. (H) Scanpy clustering (Leiden) and UMAP projection from UMAP space, with different seeds for UMAP projection.



Supplemental Figure 16: Comparison of KNN and SNN graphs. (A) KNN graph (k=3) constructed with the approximate Annoy algorithm (directedness not shown). Black = incorrectly added edges; Green = incorrectly omitted edges. (B) SNN graph with Seurat's pruning method. An example hub node is marked with *. An example peripheral node is marked with **. Note: the SNN graph with Scanpy's method is simply the undirected KNN graph.


Seurat v5.0 (default)
PCA Projection
FindNeighbors
(Annoy, RcppAnnoy)
seu@graphs$RNA_nn
KNN
seu@graphs$RNA_snn
SNN
FindClusters (Louvain)
Idents(seu)
Clusters
SNN(i,j) = 1 if jaccard(KNN(i), KNN(j)) > prune.SNN else 0
RunUMAP (uwot)
KNN
<4096 cells → exact (FNN)
≥4096 cells → approximate (Annoy, RcppAnnoy)
seu$umap@cell.embeddings
UMAP
Scanpy v1.9 (default)
PCA Projection
sc.pp.neighbors
(UMAP, umap-learn)
adata.obsp['distances']
KNN
SNN(i,j) = 1 if
i ∈ KNN(j) or
j ∈ KNN(i) else 0
adata.obsp['connectivities']
SNN
sc.tl.leiden
(Leiden, leidanalg)
adata.obs['leiden']
Clusters
<4096 cells or (<8192 cells and metric == euclidian) → exact (sklearn)
≥4096 cells or (≥8192 cells and metric == euclidian) → approximate
(NNDescent, umap-learn via PyNNDescent)
sc.tl.umap
(umap-learn)
adata.obsm["X_umap"]
UMAP

Supplemental Figure 17: Schematic of protocol describing generation and downstream use of KNN and SNN graphs. Top is for Seurat, bottom is for Scanpy. KNN(i) refers to the full set of k-nearest neighbors for cell i. Bold = function/method name; Italics = outside package utilized.

Seurat v5.0.2	Filtering	Normalization	HVG selection	Regression, Scaling	PCA	SNN	Clustering	UMAP	DE
Function nam	CreateSeuratObject	NormalizeData	FindVariableFeatures	ScaleData	RunPCA	FindNeighbors	FindClusters	RunUMAP	FindAllMarkers
Defaults	**min.cells = 0** **min.features = 0**	*scale.factor = 10000* *normalization.method = "LogNormalize"*	selection.method = "vst" **loess.span = 0.3** *clip.max = "auto"* *mean.function = FastExpMean* *dispersion.function = FastLogVMR* **num.bin = 20** *binning.method = "equal_width"* nfeatures = 2000 mean.cutoff = c(0.1, 8) dispersion.cutoff = c(1, Inf)	**vars.to.regress = NULL** *latent.data = NULL* *split.by = NULL* *model.use = "linear"* *use.umi = FALSE* *do.scale = TRUE* **do.center = TRUE** scale.max = 10	**npcs = 50** *rev.pca = FALSE* *weight.by.var = TRUE* *approx = TRUE* **features = NULL (HVGs)**	*reduction = "pca"* dims = 1:10 k.param = 20 *prune.SNN = 1/15* *nn.method = "annoy"* *n.trees = 50* **annoy.metric = "euclidean"** nn.eps = 0 *l2.norm = FALSE*	*modularity.fxn = 1* *initial.membership = NULL* *node.sizes = NULL* resolution = 0.8 method = "matrix" algorithm = 1 (original Louvain) *n.start = 10* n.iter = 10 *group.singletons = TRUE*	umap.method = "uwot" *n.neighbors = 30L* **n.components = 2L** *metric = "cosine"* **n.epochs = NULL** **learning.rate = 1** min.dist = 0.3 **spread = 1** *set.op.mix.ratio = 1* *local.connectivity = 1L* **repulsion.strength = 1** **negative.sample.rate = 5** a = NULL b = NULL dens.lambda = 2 dens.frac = 0.3 dens.var.shift = 0.1	*logfc.threshold = 0.1*[a] test.use = "wilcox" *min.pct = 0.01*[b] *min.diff.pct = -Inf* *only.pos = FALSE* *max.cells.per.ident = Inf* *latent.vars = NULL* *min.cells.feature = 3* *min.cells.group = 3* *mean.fxn = NULL* *return.thresh = 0.01* *base = 2*
(non-default) Arguments to match Scanpy defaults			selection.method = "mean.var.plot" mean.cutoff = c(0.0125, 3) dispersion.cutoff = c(0.5, Inf)	vars.to.regress = c("nCount_RNA", "pct_mt") scale.max = Inf		k.param = 15 dims = *total_pcs_used*	algorithm = "leiden" resolution = 1.0	umap.method = "umap-learn" min.dist = 0.5	logfc.threshold= 0 min.pct = 0 return.thresh = 1.0 min.cells.group = 1

[a]default changed from Seurat v4 (was logfc.threshold = 0.25)
[b]default changed from Seurat v4 (was min.pct = 0.1)

Supplemental Table 1: Seurat function details for scRNA-seq pipeline agreement with Scanpy. Bold = parameter in agreement by default; italics = no analogous paremeter in Scanpy; green = equivalent by default; yellow = equivalent with matched arguments; orange = partially incompatible, partially equivalent with matched arguments (depending on algorithm choice); red = incompatible

Scanpy v1.9.5	Filtering	Normalization	HVG selection	Regression, Scaling	PCA	SNN	Clustering	UMAP	DE
Function name	sc.pp.filter_cells, sc.pp.filter_genes	sc.pp.normalize_total	sc.pp.highly_variable_genes	sc.pp.regress_out, sc.pp.scale	sc.tl.pca	sc.pp.neighbors	sc.tl.leiden, sc.tl.louvain	sc.tl.umap	sc.tl.rank_genes_groups
Defaults	**min_cells = None** **min_genes = None** *max_cells = None* *max_genes = None* *min_counts = None (genes)* *min_counts = None (cells)* *max_counts = None (genes)* *max_counts = None (cells)*	*target_sum = None* *exclude_highly_expressed = False* *max_fraction = 0.05*	n_top_genes = None min_mean = 0.0125 max_mean = 3 min_disp = 0.5 max_disp = inf **span = 0.3** **n_bins = 20** flavor = "seurat"	**zero_center = True** max_value = None	**n_comps = 50** or 1 – minimum dimension size zero_center = True *svd_solver = "arpack"* **use_highly_variable = None (HVGs)**	n_neighbors = 15 n_pcs = None *use_rep = None* *knn = True* *method = "umap"* **metric = "euclidian"**	resolution = 1 *restrict_to = None* *directed = True* *use_weights = True* n_iterations = -1 *partition_type = None* *flavor = "vtraag" (louvain)*	min_dist = 0.5 **spread = 1.0** *n_components = 2* **maxiter = None** **alpha = 1.0** **gamma = 1** **negative_sample_rate = 5** *init_pos = "spectral"* **a = None** **b = None** method = "umap"	*use_raw = None* *reference = "rest"* n_genes = None rankby_abs = False pts = False method = None corr_method = 'benjamini-hochberg' *tie_correct = False*
(non-default) Arguments to match Seurat defaults			n_top_genes = 2000 flavor = "seurat_v3"	Omit sc.pp.regress_out max_value = 10	zero_center = False	n_neighbors = 20 n_pcs = 10 use_rep = 'X_pca'	function sc.tl.louvain instead of sc.tl.leiden resolution = 0.8 n_iterations = 10	min_dist = 0.3	use_raw=True pts = True method = "wilcoxon" corr_method = "bonferroni" tie_correct = True manual filtering (abs(log_fc) ≥ 0.1, pts ≥ 0.01, pts_rest ≥ 0.01 p_value < 0.01)

Supplemental Table 2: Scanpy function details for scRNA-seq pipeline agreement with Seurat. Bold = parameter in agreement by default; italics = no analogous paremeter in Seurat; green = equivalent by default; yellow = equivalent with matched arguments; orange = partially incompatible, partially equivalent with matched arguments (depending on algorithm choice); red = incompatible