

SA01 - Initial clustering, gene/UMI cut-off decision

Stefan Siebert

October 25, 2018

Summary

We set out to elucidate cell state specific gene and UMI metrics to make an informed decision on suitable gene and UMI cut-offs for the final data set. Drop-seq reads from 15 libraries were mapped to a de novo transcriptome reference for *Hydra vulgaris* AEP. After an initial permissive cell QC (genes per cell cut-offs of > 200 and < 8k and UMIs per cell cut-offs of > 400 < 70k) we perform graph-based clustering of the cells using Seurat (1). Clusters are annotated using published gene expression patterns. Metrics for cell states are calculated and further explored.

Preliminaries

```
library(Seurat)
library(dplyr)
library(Matrix)
library(gtable)
library(grid)
library(gridExtra)
library(rlang)

# Function to find the full ID for gene of interest Gene ids have are
# of the format txxxapl/swiss_prot_id_best_hit
hFind <- function(x) {
  return(ds.ds@data@Dimnames[[1]][grep(x, ds.ds@data@Dimnames[[1]], ignore.case = T)])
}

# We assume a folder 'objects' in the markdown directory that contains
# our raw count object and all Seurat objects
```

Load filtered data

We load `GSE121617_Hydra_DS_transcriptome_UMICounts.txt`, which is a *genes X cell* data.frame of unnormalized, unlogged transcripts detected per gene per cell.

```
# Load UMI counts
ds.counts <- read.table("objects/GSE121617_Hydra_DS_transcriptome_UMICounts.txt",
  sep = "\t", check.names = FALSE, header = TRUE)
ds.counts <- as.matrix(ds.counts)
ds.counts <- as(ds.counts, "sparseMatrix")
```

Create Seurat object

```
# Keep all genes expressed in >= 3 cells, keep all cells with >= 200
# genes
ds.ds <- CreateSeuratObject(raw.data = ds.counts, min.cells = 3, min.genes = 200,
  project = "Hydra")

# Delete original data
rm(ds.counts)
```

Basic QC

The two *Hydra* mitochondrial genomes were added to the transcriptome reference after removal of mitochondrial transcripts. This allows for calculating percentage of mitochondrial reads for each cell.

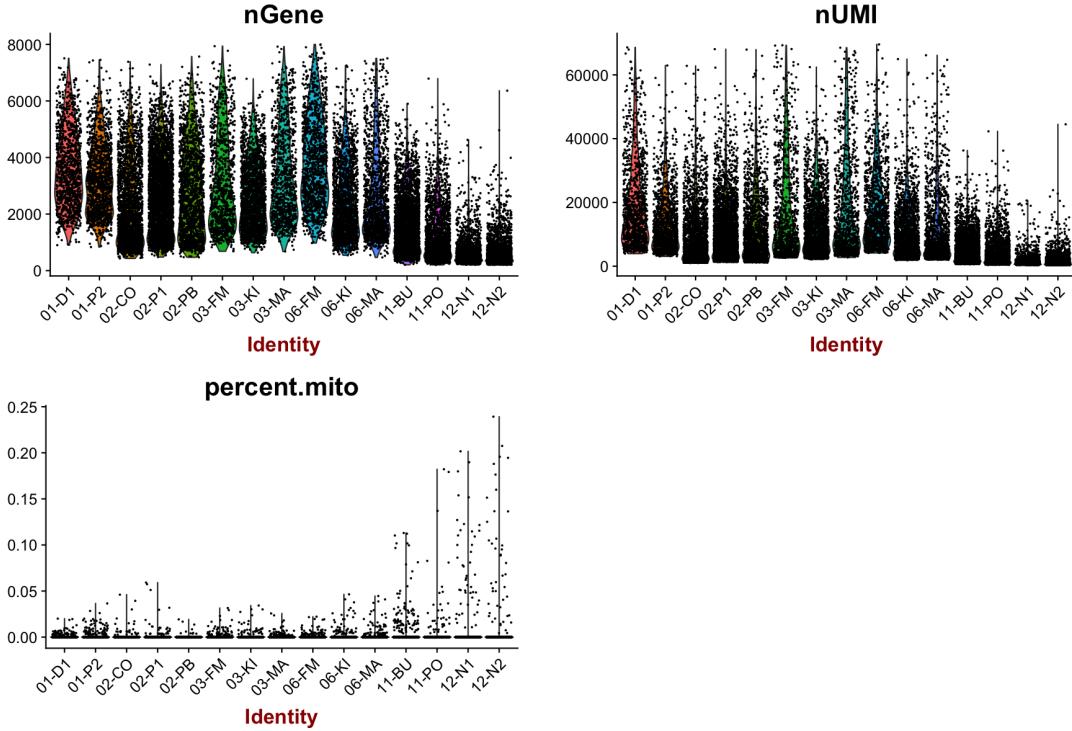


Figure 1: Number of genes, UMIs and percentage of mitochondrial reads per cell across libraries.

```
# Calculate percentage of mitochondrial reads per cell
mito.genes <- grep(pattern = "MT_", x = rownames(x = ds.ds@data), value = TRUE)
percent.mito <- Matrix::colSums(ds.ds@raw.data[mito.genes, ])/Matrix::colSums(ds.ds@raw.data) * 100

# Add percent.mito column to object@data.info
ds.ds <- AddMetaData(ds.ds, percent.mito, "percent.mito")
```

We evaluate genes/cell, UMIs/cell and percentage of mitochondrial reads across treatments (Fig. 1). Two different types of beads were used in the experiments. Libraries 01-06 were generated using original Drop-seq beads (Beaded Beads SeqB; ChemGenes Corp., Wilmington, MA, USA) (2)). Libraries 11-12 were generated using R&D beads (LGC Biosearch Technologies, Petaluma, CA, USA) that had a 12bp UMI as the only modification to the linker. A longer sequencing read 1 yielded improved sequence quality scores on the Illumina NextSeq. Biosearch beads however generated lower gene and UMI numbers per cell for comparable cell suspensions (Figs. 1, 2), e.g. libraries 06-KI and 11-PO were both created using suspensions from whole *Hydra*. The highest percentages of mitochondrial reads were observed in FACS sorted neuronal libraries 12-N1 and 12-N2 (Figs. 1, 2).

```
# Fig. qcstats
VlnPlot(object = ds.ds, c("nGene", "nUMI", "percent.mito"), group.by = "orig.ident",
         nCol = 2, x.lab.rot = TRUE, point.size.use = 0.2)
```

High levels of mitochondrial reads may indicate that a cell was stressed or dying. We exclude cells that have > 5% of mitochondrial reads and calculate post filtering stats for cells from individual libraries (Fig. 2).

```
# Calculate library stats
mylist <- list() #create an empty list

# Calculate library stats
for (i in levels(ds.ds@meta.data$orig.ident)) {
  vec <- numeric(3)
  s <- SubsetData(object = ds.ds, ident.use = i)
  all <- length(s@meta.data$nGene)
  vec[1] <- as.numeric(length(s@meta.data$nGene))
  s <- FilterCells(object = s, subset.names = c("percent.mito"), low.thresholds = c(-Inf),
```

	cell# (pre)	% high mito	cell# (post)	genes (post)	UMIs (post)
01-D1	1086	0	1086	3486	15874
01-P2	1299	0	1299	3123	10618
02-CO	2395	0	2395	1947	5242
02-P1	3498	0.09	3495	2386	7061
02-PB	1599	0	1599	2634	8206
03-FM	965	0	965	2747	11092
03-KI	1997	0	1997	2474	9990
03-MA	984	0	984	2979	11762
06-FM	1188	0	1188	3792	14300
06-KI	1997	0	1997	2087	7349
06-MA	1494	0	1494	2104	6372
11-BU	3299	0.39	3286	1344	3282
11-PO	2179	0.28	2173	936	2263
12-N1	1557	1.86	1528	507	906
12-N2	2451	1.06	2425	477	895

Figure 2: Percentage of cells with high mitochondrial read ratios across libraries. Cell numbers pre and post filtering. Fraction of mitochondrial reads. Median genes and UMIs per cell post filtering using permissive gene and UMI cut-offs.

```

    high.thresholds = c(0.05))
vec[3] <- length(s@meta.data$nGene)
low_mito <- length(s@meta.data$nGene)
percent <- (all - low_mito)/all * 100
vec[2] <- round(percent, digits = 2)
vec[4] <- round(median(s@meta.data$nGene), digits = 0)
vec[5] <- round(median(s@meta.data$nUMI), digits = 0)
mylist[[i]] <- vec
}

df <- do.call("rbind", mylist) #combine all vectors into a matrix
df <- as.data.frame(df)
colnames(df) <- c("cell# (pre)", "% high mito", "cell# (post)", "genes (post)",
"UMIs (post)")

tt <- ttheme_default(core = list(fg_params = list(hjust = 1, x = 0.9)),
rowhead = list(fg_params = list(hjust = 1, x = 0.95)))

g <- tableGrob(df, theme = tt)
g <- gtable_add_grob(g, grobs = rectGrob(gp = gpar(fill = NA, lwd = 2)),
t = 2, b = nrow(g), l = 1, r = ncol(g))
g <- gtable_add_grob(g, grobs = rectGrob(gp = gpar(fill = NA, lwd = 2)),
t = 1, l = 1, r = ncol(g))

grid.draw(g)

# Exclude cells with high percentage of mitochondrial reads
ds.ds <- FilterCells(object = ds.ds, subset.names = c("percent.mito"),
low.thresholds = c(-Inf), high.thresholds = c(0.05))

```

Pre-clustering workflow

We use Seurat to normalize and scale the data and to identify genes that vary more than expected for their expression level (Fig. 3).

```

# Normalize
ds.ds <- NormalizeData(object = ds.ds, normalization.method = "LogNormalize",
scale.factor = 10000)

```

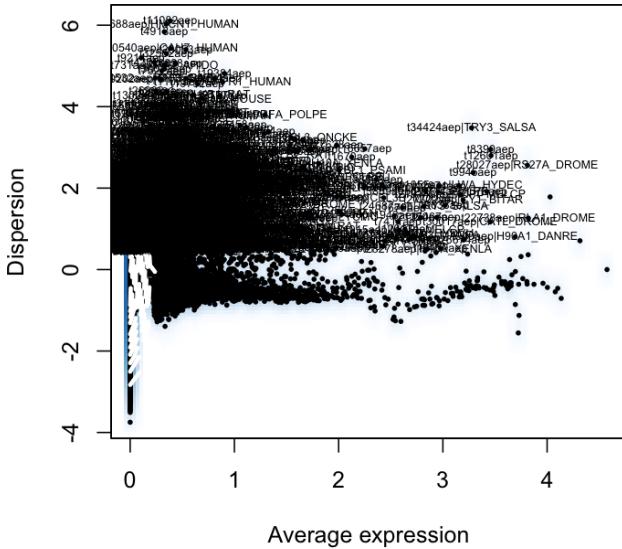


Figure 3: Variable genes using a dispersion cut-off of 0.5 and an expression cut-off of 0.05.

```
# Fig. vargenes, identify highly variable genes
ds.ds <- FindVariableGenes(object = ds.ds, mean.function = ExpMean, dispersion.function = LogVMR,
  x.low.cutoff = 0.05, x.high.cutoff = 4, y.cutoff = 0.5)

# Scale
ds.ds <- ScaleData(object = ds.ds)
```

Perform linear dimensional reduction

In preparation for graph-based clustering we perform PCA on the scaled data using genes in object@var.genes. 2360 genes are identified as variable given the selected cut-offs for dispersion and expression level.

```
# PCA on highly variable genes
ds.ds <- RunPCA(object = ds.ds, pc.genes = ds.ds@var.genes, pcs.compute = 40,
  do.print = TRUE, pcs.print = 1:5)
```

We score each gene in the dataset (including genes not included in the PCA) based on their correlation with the calculated components.

```
# Project PCA
ds.ds <- ProjectPCA(object = ds.ds)
```

Determine PCs to be included in downstream analyses

We perform the Jackstraw test as implemented in Seurat to identify “significant” principal components to be used in the cell clustering. In addition, we approximate the amount of variance encoded by each PC using the function PCELbowPlot() (Fig. 4).

```
# Determine statistically significant PCs
ds.ds <- JackStraw(object = ds.ds, num.pc = 40, num.replicate = 100, do.print = FALSE)
JackStrawPlot(object = ds.ds, PCs = 1:40)

# Fig., approximate amount of variance encoded by each PC Look at a
# plot of the standard deviations of the principle components and draw
# cut-off where there is a clear elbow in the graph
PCELbowPlot(object = ds.ds, num.pc = 40)
```

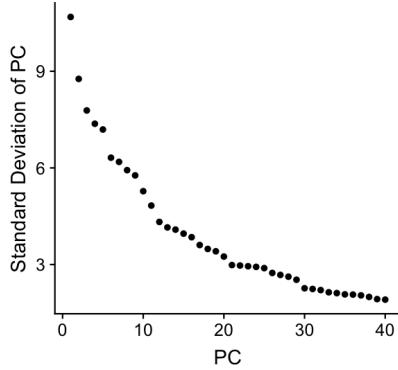


Figure 4: Plot of the standard deviations of the principle components.

Community detection clustering

At this stage we cluster the cells to obtain broad insights into the captured cell states. We consider PCs 1:19 in this initial cell clustering which results in cell lineage separation and biologically meaningful clustering results. The JackStraw test finds > 40 PCs to be statistically significant. Considering a higher number of principal components reveals additional variability within cell populations including batch effects which we evaluate in detail further downstream.

```
# Find cluster
ds.ds <- FindClusters(object = ds.ds, reduction.type = "pca", dims.use = 1:19,
force.recalc = TRUE, resolution = 1.5, print.output = 0)
```

t-SNE embedding

We apply t-SNE dimensionality reduction for visualization purposes. 44 clusters are recovered in our original analysis using a resolution parameter of 1.5 (Fig. 5). t-SNE dimensionality reduction is not deterministic and resulting plots will differ.

```
# Run t-SNE
ds.ds <- RunTSNE(object = ds.ds, dims.use = c(1:19), do.fast = T)

# Save object saveRDS('objects/ds.ds.g200_8k_U400_70k_PC1_19.rds')

# Since t-SNE is not deterministic we here load the object of our
# original analysis
ds.ds <- readRDS("objects/ds.ds.g200_8k_U400_70k_PC1_19.rds")

# Fig. tsne
TSNEPlot(object = ds.ds, group.by = "res.1.5", do.return = T, do.label = T,
no.legend = TRUE, pt.size = 0.5) + ggtitle("Resolution 1.5")
```

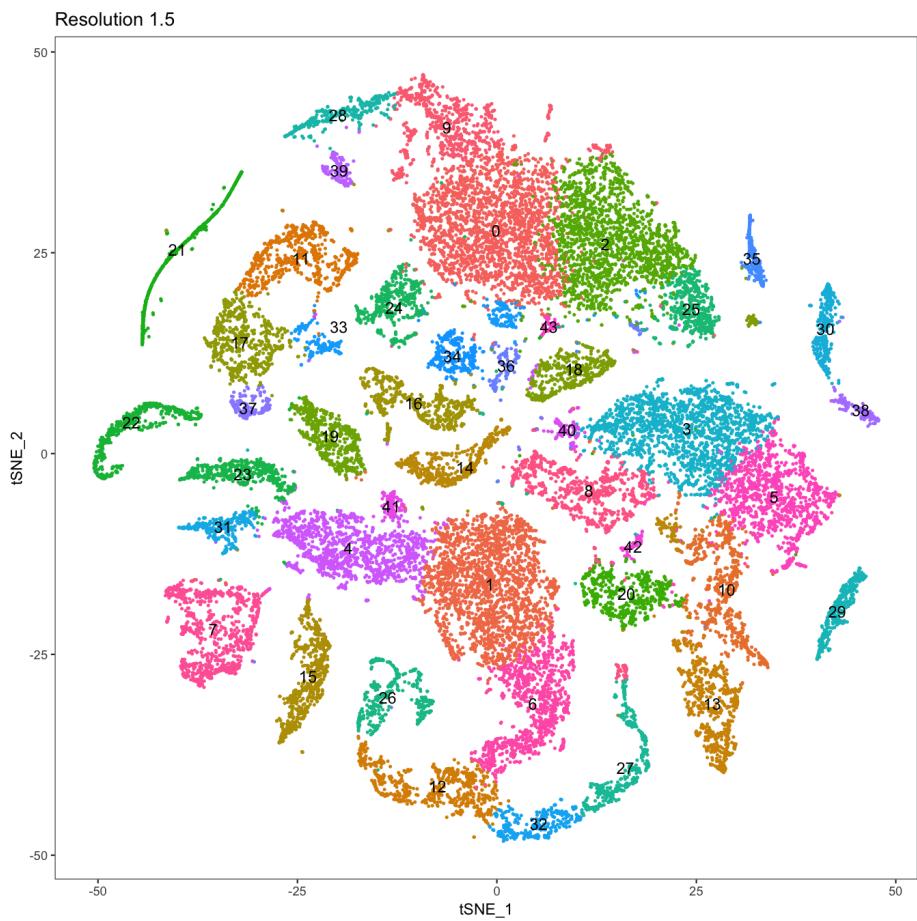


Figure 5: t-SNE plot for all cells after applying permissive gene and UMI cut-offs.

Cluster annotation

We use selected markers to get insights into cluster identity and visualize expression on the t-SNE representation (Fig. 6). We label the cluster according to the reported expression domain of the selected marker. Multiple markers were used to refine the annotation (Fig. 7). Only a selection of markers is presented here.

```
# Annotate t-SNE Selected genes used for annotation
gene.names <- c(hFind("t31074aep"), hFind("t14194aep"), hFind("t25396aep"),
  hFind("t16043aep"), hFind("t4498aep"), hFind("t16456aep"), hFind("t11407aep"),
  hFind("t3974aep"), hFind("t8678aep"), hFind("t12596aep"), hFind("t7059aep"),
  hFind("t13480aep"), hFind("t23176aep"), hFind("t11117aep"), hFind("t11585aep"))

# Update gene names
new.names <- c("armininia1a", "HyWnt3", "CnNK-2", "PPOD1", "ks1", "HyAlx",
  "Cnnos1", "ELAV2 (t3974)", "HyDkk1/2/4 A", "HyTSR1", "MUC2 (t7059)",
  "nematoglectin B", "nematocilin A", "periculin1a", "H2BL1 (t11585)")

# Function to annotate/ rename genes
update.names <- function(gene.names, new.names) {
  for (i in 1:length(gene.names)) {
    rownames(ds.ds@data)[which(rownames(ds.ds@data) == gene.names[i])] <- new.names[i]
  }
}

# Annotate
update.names(gene.names, new.names)

# Plot with tsne
p1 <- TSNEPlot(object = ds.ds, group.by = "res.1.5", do.label = T, label.size = 5,
  pt.size = 0.5, cex.names = 6, no.legend = TRUE, do.return = TRUE)
p2 <- FeaturePlot(ds.ds, c("armininia1a", "HyWnt3", "CnNK-2", "PPOD1", "ks1",
  "HyAlx", "Cnnos1", "ELAV2 (t3974)", "HyDkk1/2/4 A", "HyTSR1", "MUC2 (t7059)",
  "nematoglectin B", "nematocilin A", "periculin1a", "H2BL1 (t11585)", cols.use = c("grey", "blue"), do.return = TRUE)

plotlist <- prepend(p2, list(p1))
plot_grid(plotlist = plotlist, labels = "AUTO", label_size = 20, align = "h",
  ncol = 4)

# Annotate clusters Store cluster numbering
ds.ds <- StashIdent(object = ds.ds, save.name = "cluster_numbering")
ds.ds <- SetAllIdent(ds.ds, "res.1.5")

current.cluster.ids <- as.character(0:43)

# Run this to restore original cluster numbering
ds.ds <- SetAllIdent(object = ds.ds, id = "cluster_numbering")

# Cluster identities
cluster.names <- c("enEp_SC1", "i_SC", "enEp_SC2", "ecEp_SC1", "i_gc_nc_prog",
  "ecEp_SC2", "i_nb1", "i_smgc", "ecEp_nem1(pg)", "enEp_foot", "ecEp_head",
  "i_gmgc", "i_nb2", "ecEp_bat(mp)", "ecEp_bd", "i_nc1", "i_nem", "i_zmg1",
  "i_nc2", "i_fmg1", "ecEp_nem2(id)", "i_mgl1", "i_fmg12", "i_nc3", "enEp_head",
  "enEp_SC3", "i_nb3", "i_nb5", "enEp_tent", "i_nc4", "i_nc5", "i_nc6",
  "i_nb4", "db3", "enEp_nem1(pd)", "i_nc7", "enEp_nem2(pd)", "i_zmg2",
  "i_nc8", "enEp_tent(pd)", "db1", "i_gc_prog", "ecEp_nem3(pg)", "db2")

# Update names in Seurat object
ds.ds@ident <- plyr::mapvalues(x = ds.ds@ident, from = current.cluster.ids,
  to = cluster.names)

# Fig. annotated tsne
TSNEPlot(object = ds.ds, do.return = T, do.label = T, no.legend = TRUE,
  pt.size = 0.5) + ggtitle("g>200<8k genes, UMI>400<70k, 27,911 cells")
```

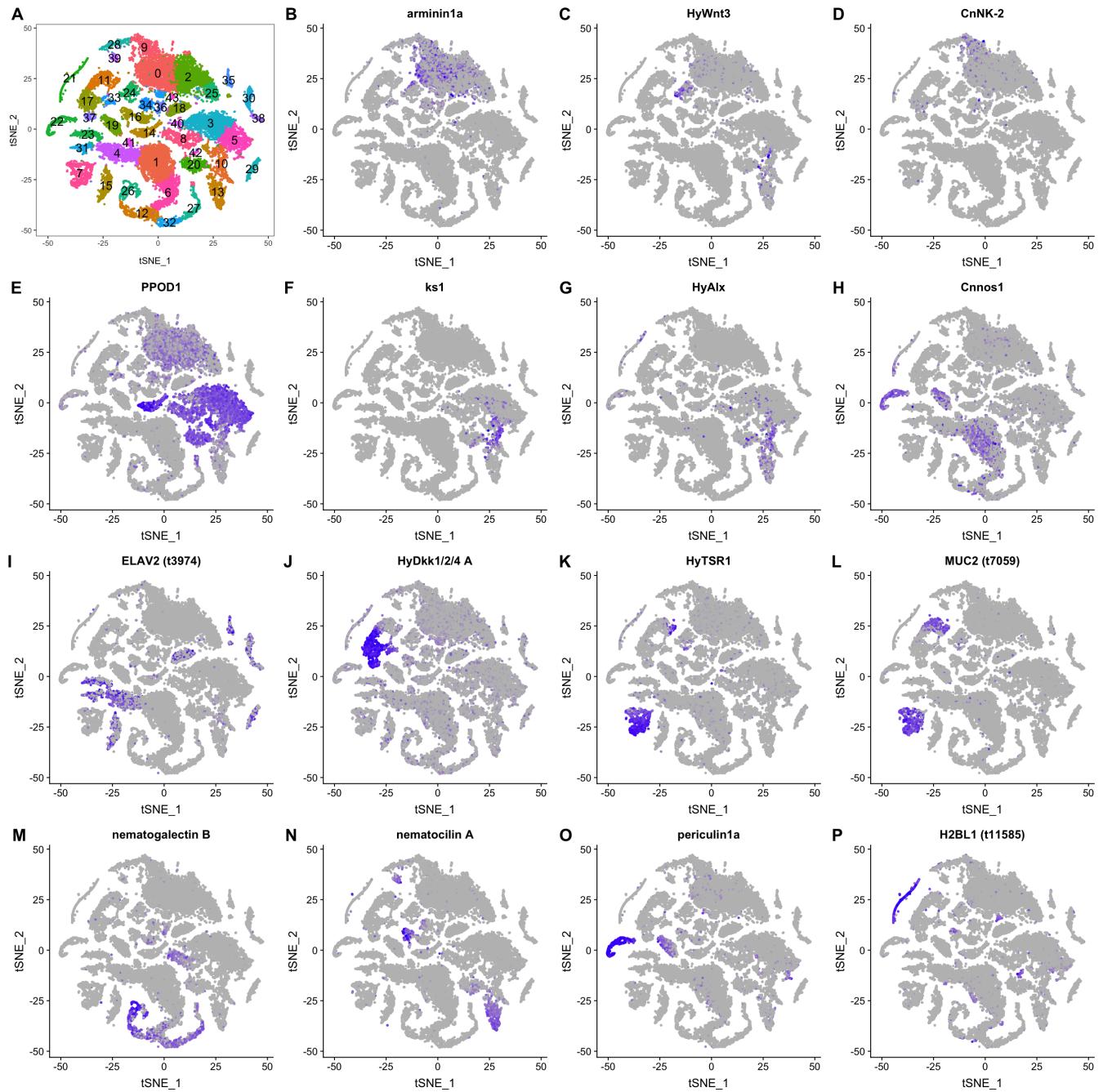


Figure 6: Selected markers used for cluster annotation. A) t-SNE plot for permissive data set. B) endoderm - *arminin1a* (3). C) endoderm/ectoderm hypostome - *Wnt3* (4). D) endoderm foot/peduncle - *CnNK-2* (5). E) ectoderm/endoderm - *PPOD1* (6). F) ectoderm head - *ks1* (7). G) ectoderm tentacle - *HyAix* (8). H) multipotent i-cells/progenitors/male and female germline - *Cnnos1* (9). I) neuron progenitor/neurons - *ELAV2* (t3974, this study). J) zymogen gland cell - *HyDkk1/2/4-A* (10, 11). K) mucous gland cells - *HyTSR1* (12). L) mucous gland cell - *MUC2* (t7059). M) nematogenesis/doublets - *nematoglectin B* (13). N) differentiated nematocytes/battery cell - *nematocillin A* (14). O) female germline - *periculin1a* (15). P) male germline - histone *H2BL1* (t11585, this study).

$g > 200 < 8k$ genes, $UMI > 400 < 70k$, 27,911 cells

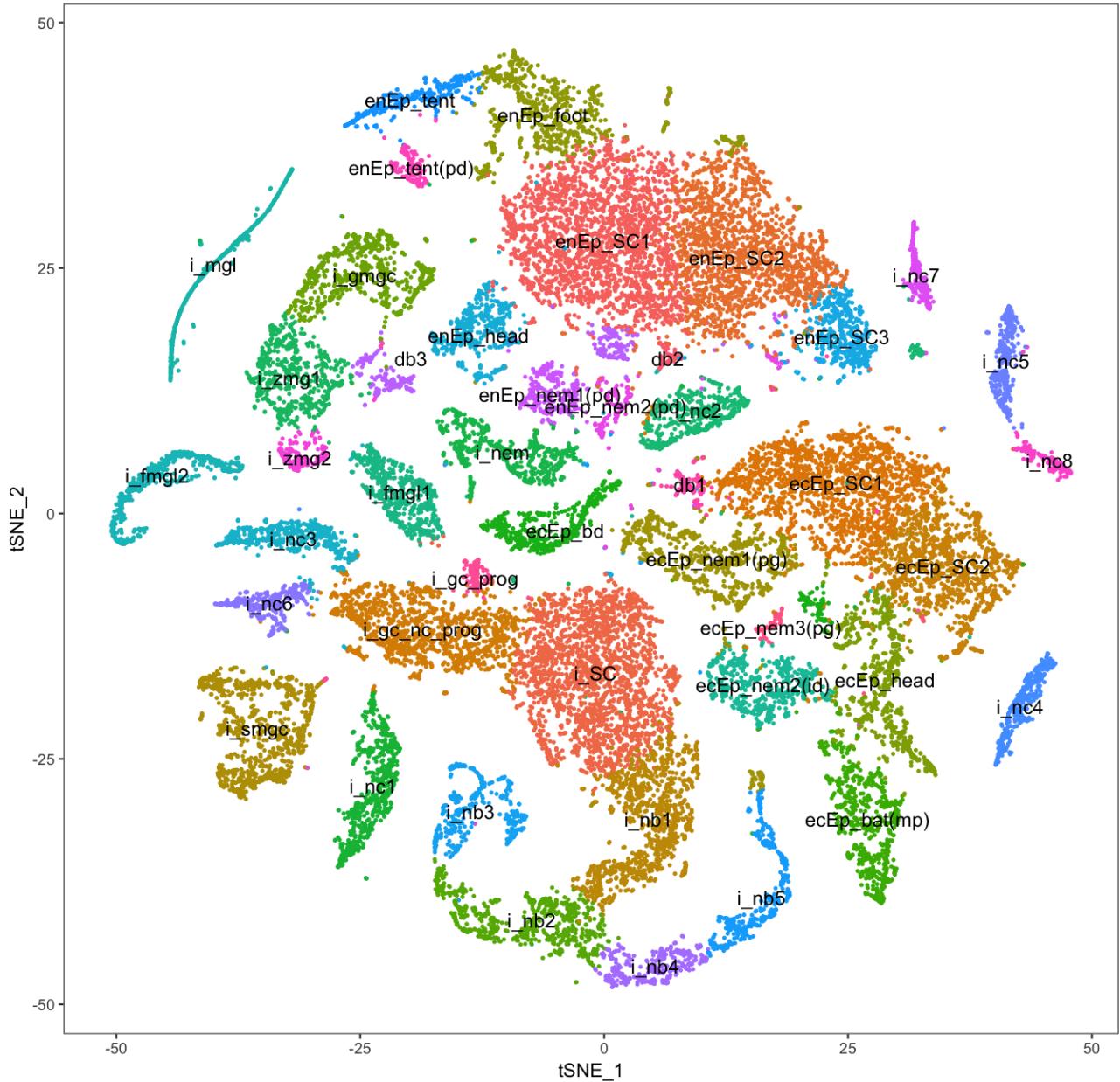


Figure 7: Annotated t-SNE plot for the permissive dataset. bat: battery cell, db: doublet, ecEP: ectodermal epithelial cell, enEP: endodermal epithelial cell, fmgl: female germ-line, gc: gland cell, gmgc: granular mucous gland cell, i: cell of the interstitial lineage, id: integration doublet, mgl: male germline, mp: multiplet, nb: nematoblast, nc: neuronal cell, nem: nematocyte, pd: suspected phagocytosis doublet, prog: progenitor, SC: stem cell, smgc: spumous mucous gland cell, tent: tentacle, zmg: zymogen gland cell.

Cluster annotation identifies biological multiplets

Hydra has a specialized cell that is localized in the ectoderm of the tentacles - the battery cell (16, 17) (Fig. 8). This epithelial cell houses multiple nematocytes and a neuron (16, 17) (Fig. 8 D,G). Our dissociation strategy does not allow for separation of the host cell and the mounted cells and this type of association shows as a hybrid transcriptome in the data. Cells in clusters composed of battery cell multiplets are positive for epithelial, nematocyte and neuronal markers, and are labeled (mp) in the annotated t-SNE plot. In addition, mature nematocytes can be found mounted or integrated in ectodermal epithelial cells along the body column and neurons have also been found integrated within ectodermal cells outside the tentacles (18, 19). We refer to a physical association of a neuron/nematocyte and an epithelial cell outside the tentacles that naturally occurs in homeostatic *Hydra* as integration doublet (id).

We find unexpected colocalization of nematocyte gene expression with endodermal epithelial gene expression, e.g. cells of cluster 36 (enEP_nem2(pd)) show expression for both endodermal markers and the nematoblast marker *nematogalectin B* (Fig. 6 M). Except for a single publication (20), endodermal localization of nematocytes has not been established in *Hydra*. We attribute these doublet transcriptional signatures, in part, to phagocytotic activity of epithelial cells. Phagocytotic activity of epithelial cells has been reported on multiple occasions (20, 21). Phagocytosis may occur across the mesoglea via phagocytotic processes even in homeostatic *Hydra* and evidence is observable as cellular remains inside unbroken epithelial cells separated by maceration in wild type *Hydra* (21). Using transgenic lines expressing GFP in specific interstitial cell populations we are able to find evidence in support for the hypothesis that neighbouring cells get taken up during the dissociation procedure (Fig. 8 A). Phagocytotic uptake occurs within 5 to 30min post challenge and thus likely occurs before the animals are fully dissociated (22). The uptake is not restricted to entire cells but also includes blebs that pinch off from epithelial cells during dissociation; we demonstrated this by using cells that express cytosolic fluorescent proteins (not shown). This could potentially lead to spurious expression of marker genes from different lineages in epithelial cells of the ectoderm or endoderm. Phagocytotic uptake of cells or blebs has also been demonstrated in reaggregation experiments (23). We interpret co-expression of nematoblast/nematocyte markers and endodermal markers in part as a result of these processes. This doublet type is labeled as suspected phagocytosis doublet (pd) in the annotated t-SNE plot. Phagocytotic uptake of nematoblasts by epithelial cells is obvious in the t-SNE representation (Fig. 5). Cluster i_nb5 are late stage nematoblasts. The terminus of this trajectory is formed by cells positive for *PPOD1* expression, an epithelial marker (compare to Fig. 6 E), suggesting that the terminal group of cells in this trajectory represent doublets composed of nematoblasts and ectodermal epithelial cells. A clear distinction between the mp, id and pd categories is in many cases not possible.

Neuronal multiplets and doublets resulting from integration or phagocytotic uptake could be demonstrated in fluorescence-activated sorted cell (FACS) populations (Fig. 9). The line (eGreen) used for this experiment expresses GFP (act::GFP) predominantly in the neuronal trajectory and mature neurons as a result of a random integration event after zygote microinjection. Using FACS, we were able to collect both single neuronal cells (Fig. 9 B) and larger GFP-positive cells that we determined to be multiplets composed of GFP-positive cells residing within epithelial cells (Fig. 9 C). Ectodermal cells of the body column or endodermal cells are suggested as host cells in cases where co-integrated nematocytes are absent (see also (Fig. 8 H, I)). The presence of multiple biological doublet categories as well as technical doublets (e.g. dissociation doublets and expected drop-seq doublets) imposes analysis challenges. We therefore did not perform global doublet exclusion at an early stage but rather evaluated doublets at each stage of the downstream analyses and applied strategies to appropriately deal with these challenges.

Cells and UMIs across cell states, bead evaluation

We determine gene and UMI metrics for each annotated cell state in this initial clustering. The overall statistics are summarized in Fig. 10. The lowest gene and UMI numbers are observed for differentiated neurons. Cells that were identified as female germline (nurse cells) have the highest gene and UMI numbers. Zymogen gland cells (i_zmg1, i_zmg2) stand out as cells with relatively high UMI to gene ratios (Fig. 10).

Since choice of beads affected the cell metrics, we break down the metrics by beads (Fig. 11). We compare metrics for cells generated with ChemGenes beads (CG, 11 libraries, whole animal dissociations) to all cells that were generated using Biosearch beads (BS, 4 libraries). We further break down Biosearch beads into cells that were generated from whole animal dissociations (wBS, 2 libraries, to be compared to ChemGenes bead metrics) and cells that were enriched for neuronal cells using FACS (nBS, 2 libraries). Both bead

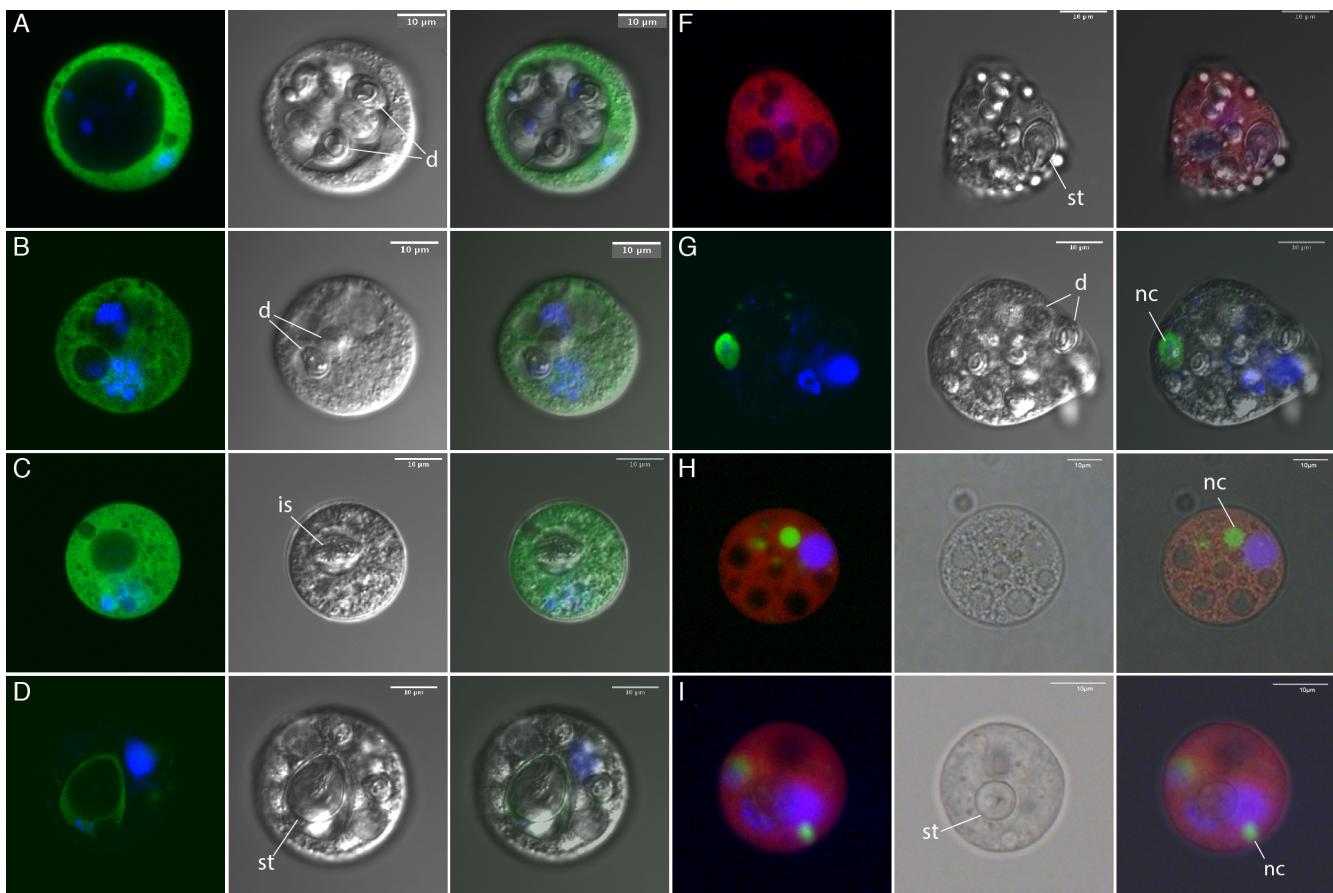


Figure 8: Documentation of live cells after tissue dissociation reveals cell multiplets. Multiple transgenic strains expressing fluorescent proteins were used to demonstrate the existence of biological multiplets (e.g. battery cells), viability of cells within the host cells, and the occurrence of phagocytosis. For all panels: 1st image fluorescence, 2nd image bright field, 3rd image overlay. A-C,F) Line expressing GFP in ectodermal epithelial cells and RFP in endodermal epithelial cells. D,G) Line (eGreen) expressing GFP predominantly in the neuronal lineage and scattered in nematocytes. H,I) Cross of lines neuro::GFP and eGreen expressing GFP in ectodermal neurons and RFP in epithelial cells (Lines courtesy of Rob Steele, Toshitaka Fujisawa, Xiaoming Zhang and Yukihiko Noro). A-D) Ectodermal epithelial cells containing a single or multiple nematocytes of one or multiple kinds. A) Ectodermal epithelial cell containing multiple nematocytes with desmonemes. The nematocytes appear to be contained in a vacuole like compartment that may be indicative of phagocytotic uptake in the course of the dissociation procedure. B) Ectodermal epithelial cell containing two nematocytes with desmonemes. C) Ectodermal epithelial cell containing a single nematocyte with isorhiza nematocyst. D) Nematocyte with stenotele and GFP positive cytoplasm within an ectodermal epithelial battery cell. F) Endodermal epithelial cell containing a stenotele. G) Ectodermal epithelial battery cell containing a GFP positive neuron. H) Endodermal epithelial cell (as indicated by the presence of vacuoles) containing GFP, with transgenic neurons as the sole possible source for GFP. I) Ectodermal epithelial cell containing a nematocyte (stenotele) and GFP positive structures, with transgenic neurons as the sole possible source for GFP. d: desmoneme, is: isorhiza, st: stenotele, nc: neuronal cell.

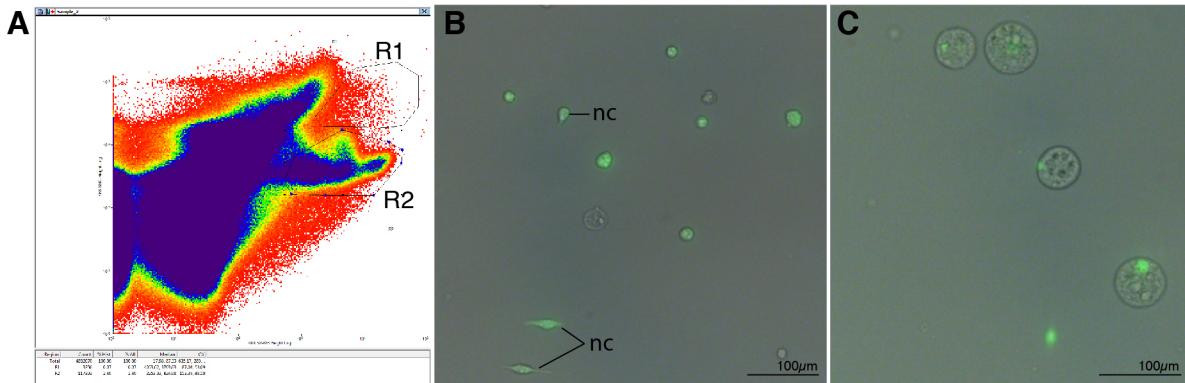


Figure 9: Pilot fluorescence-activated cell sorting (FACS) of *Hydra* cell suspensions. A) Gates implemented to collected GFP-positive cells. Both populations, R1 and R2, are GFP-positive. Population R2 (B) contained undifferentiated cells, neurons (bottom, to be labeled), and scattered nematocytes. Cells of population R2 were collected in two independent sortings and used to generate two Drop-seq libraries. C) Population R1 was characterized by lower GFP intensities and larger cell sizes. Fluorescent microscopy combined with DIC (C) revealed GFP signal within larger epithelial cells demonstrating the existence of biological multiplets likely due to integrated neurons or phagocytosis. These cell multiplets are captured when performing whole animal cell suspensions.

types generate similar distributions of gene and UMI numbers across cell states with fewer genes and UMIs detected in case of Biosearch beads.

```
# Calculate median gene and UMI numbers for all cluster
mylist <- list() #create an empty list

for (i in levels(ds.ds@ident)) {
  vec <- numeric(3) #preallocate a numeric vector
  # vec[1] <- as.numeric(i)
  s <- SubsetData(object = ds.ds, ident.use = i)
  vec[1] <- round(median(s@meta.data$nGene), digits = 0)
  vec[2] <- round(median(s@meta.data$nUMI), digits = 0)
  vec[3] <- length(s@meta.data$nGene)
  mylist[[i]] <- vec #put all vectors in the list
}
df <- do.call("rbind", mylist) #combine all vectors into a matrix
df <- as.data.frame(df)
colnames(df) <- c("medianGene", "medianUMI", "cells")

df <- df[order(df$medianGene), ]
df <- df[, c(3, 1, 2)]

# Table genes/UMIs, create table for median gene and UMI numbers for
# all states for cells from all libraries

# https://stackoverflow.com/questions/44141060/how-to-formatting-numbers-by-column-in-a-table-tablegrob
# Function to scale each column to the range
norm <- function(x) {
  apply(x, 2, function(y) {
    (y - min(y))/(max(y) - min(y))
  })
}

bluecol <- colorRamp(c("red", "yellow", "green"))(norm(df))
bluecol <- rgb(bluecol[, 1], bluecol[, 2], bluecol[, 3], max = 255)

tt <- ttheme_default(core = list(bg_params = list(fill = bluecol)))

g <- tableGrob(df, theme = tt)
g <- gtable_add_grob(g, grobs = rectGrob(gp = gpar(fill = NA, lwd = 2)),
  t = 2, b = nrow(g), l = 1, r = ncol(g))
g <- gtable_add_grob(g, grobs = rectGrob(gp = gpar(fill = NA, lwd = 2)),
```

```

t = 1, l = 1, r = ncol(g))
grid.draw(g)

# Calculate median gene and UMI numbers for all clusters by bead type

# Get library ids > levels(ds.ds@meta.data$orig.ident) [1] '01-D1'
# '01-P2' '02-C0' '02-P1' '02-PB' '03-FM' '03-KI' '03-MA' '06-FM'
# '06-KI' '06-MA' '11-BU' '11-PO' '12-N1' '12-N2'

# Get cells from ChemGenes libraries
S1 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "01-D1"]
S2 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "01-P2"]
S3 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "02-C0"]
S4 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "02-P1"]
S5 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "02-PB"]
S6 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "03-FM"]
S7 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "03-KI"]
S8 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "03-MA"]
S9 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "06-FM"]
S10 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "06-KI"]
S11 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "06-MA"]

# Combine cells from ChemGenes libraries
chem <- c(S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11)

# Calculate median gene and UMI numbers per cluster for cells from
# ChemGenes libraries
mylist <- list()

for (i in levels(ds.ds@ident)) {
  vec <- numeric(3)
  s <- SubsetData(object = ds.ds, ident.use = i)
  s <- SubsetData(object = s, cells.use = chem)
  vec[1] <- round(median(s@meta.data$nGene), digits = 0)
  vec[2] <- round(median(s@meta.data$nUMI), digits = 0)
  vec[3] <- length(s@meta.data$nGene)
  mylist[[i]] <- vec
}
cg <- do.call("rbind", mylist)
cg <- as.data.frame(cg)

colnames(cg) <- c("CGmedGene", "CGmedUMI", "CGcells")
cg$cellState <- rownames(cg)
cg <- cg[, c(4, 3, 1, 2)]

# Get cells from Biosearch libraries
S12 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "11-BU"]
S13 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "11-PO"]
S14 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "12-N1"]
S15 <- rownames(ds.ds@meta.data)[ds.ds@meta.data[, "orig.ident"] == "12-N2"]

# Combine cells from Biosearch libraries
bio <- c(S12, S13, S14, S15)

# Calculate median gene and UMI numbers per cluster for cells from
# Biosearch libraries
mylist <- list()

for (i in levels(ds.ds@ident)) {
  vec <- numeric(3)
  s <- SubsetData(object = ds.ds, ident.use = i)
  s <- SubsetData(object = s, cells.use = bio)
  vec[1] <- round(median(s@meta.data$nGene), digits = 0)
  vec[2] <- round(median(s@meta.data$nUMI), digits = 0)
  vec[3] <- length(s@meta.data$nGene)
  mylist[[i]] <- vec
}

BS <- do.call("rbind", mylist)
BS <- as.data.frame(BS)

```

	cells	medianGene	medianUMI
<i>i_nc1</i>	637	401	752
<i>i_nc3</i>	502	422	810
<i>i_nc7</i>	250	436	620
<i>i_nc6</i>	330	444	814
<i>i_nc2</i>	547	498	906
<i>i_nc5</i>	336	516	988
<i>i_nc4</i>	357	521	929
<i>i_nb3</i>	442	537	1312
<i>i_nc8</i>	175	609	1166
<i>i_nem</i>	635	953	1838
<i>enEp_SC3</i>	472	968	1976
<i>i_zmg2</i>	181	1120	7718
<i>i_gmgc</i>	689	1177	3994
<i>i_nb5</i>	419	1186	4293
<i>i_nb2</i>	676	1190	2740
<i>i_smgc</i>	870	1218	3540
<i>i_nb4</i>	320	1276	3362
<i>i_gc_nc_prog</i>	1277	1315	2981
<i>i_zmg1</i>	603	1423	9009
<i>db1</i>	144	1590	3889
<i>i_nb1</i>	1060	1676	4298
<i>enEp_tent</i>	364	1714	4837
<i>i_mgl</i>	539	1806	4094
<i>enEp_SC2</i>	1801	2077	5950
<i>ecEp_bd</i>	644	2081	6604
<i>db2</i>	56	2090	6488
<i>enEp_tent(pd)</i>	152	2142	6306
<i>i_gc_prog</i>	128	2160	6044
<i>ecEp_SC1</i>	1709	2216	6697
<i>enEp_head</i>	480	2240	6029
<i>i_SC</i>	2129	2249	7291
<i>enEp_foot</i>	795	2329	6882
<i>i_fmg1</i>	542	2678	7074
<i>db3</i>	304	3003	10638
<i>ecEp_head</i>	747	3018	10210
<i>enEp_nem1(pd)</i>	298	3032	10042
<i>enEp_nem2(pd)</i>	182	3160	10974
<i>ecEp_bat(mp)</i>	669	3269	10638
<i>ecEp_nem3(pg)</i>	82	3428	11422
<i>ecEp_nem2(id)</i>	539	3550	13313
<i>enEp_SC1</i>	2265	3756	15367
<i>ecEp_nem1(pg)</i>	860	3768	15164
<i>ecEp_SC2</i>	1184	3894	16411
<i>i_fmg2</i>	520	4760	17361

Figure 10: Median genes/UMIs per cell per state (all libraries).

```

colnames(BS) <- c("BSmedGene", "BSmedUMI", "BScells")
BS$cellState <- rownames(BS)
BS <- BS[, c(4, 3, 1, 2)]

cgBS <- merge(CG, BS, by = "cellState")
cgBS <- cgBS[order(cgBS$CGmedGene), ]
cgBS[is.na(cgBS)] <- 0

rownames(cgBS) <- cgBS$cellState
cgBS$cellState <- NULL

# Get cells from Biosearch libraries 11-, these were whole animal
# dissociations
bio <- c(S12, S13)

# Calculate median gene and UMI numbers per cluster for cells from
# Biosearch libraries 11-
mylist <- list() #create an empty list

for (i in levels(ds.ds@ident)) {
  vec <- numeric(3)
  s <- SubsetData(object = ds.ds, ident.use = i)
  s <- SubsetData(object = s, cells.use = bio)
  vec[1] <- round(median(s@meta.data$nGene), digits = 0)
  vec[2] <- round(median(s@meta.data$nUMI), digits = 0)
  vec[3] <- length(s@meta.data$nGene)
  mylist[[i]] <- vec
}

wBS <- do.call("rbind", mylist)
wBS <- as.data.frame(wBS)

colnames(wBS) <- c("wBSmedGene", "wBSmedUMI", "wBScells")
wBS$cellState <- rownames(wBS)
wBS <- wBS[, c(4, 3, 1, 2)]

cgwBS <- merge(CG, wBS, by = "cellState")
cgwBS <- cgwBS[order(cgwBS$CGmedGene), ]
cgwBS[is.na(cgwBS)] <- 0

rownames(cgwBS) <- cgwBS$cellState
cgwBS$cellState <- NULL

# get cells from Biosearch libraries 12-, these were FACS sorted
# libraries
bio <- c(S14, S15)

# calculate median gene and UMI numbers per cluster for cells from
# Biosearch libraries 12-
mylist <- list()

for (i in levels(ds.ds@ident)) {
  vec <- numeric(3)
  s <- SubsetData(object = ds.ds, ident.use = i)
  s <- SubsetData(object = s, cells.use = bio)
  vec[1] <- round(median(s@meta.data$nGene), digits = 0)
  vec[2] <- round(median(s@meta.data$nUMI), digits = 0)
  vec[3] <- length(s@meta.data$nGene)
  mylist[[i]] <- vec
}

nBS <- do.call("rbind", mylist)
nBS <- as.data.frame(nBS)

colnames(nBS) <- c("nBSmedGene", "nBSmedUMI", "nBScells")
nBS$cellState <- rownames(wBS)
nBS <- nBS[, c(4, 3, 1, 2)]

wBSnBS <- merge(wBS, nBS, by = "cellState")
wBSnBS <- wBSnBS[order(wBSnBS$wBSmedGene), ]
wBSnBS[is.na(wBSnBS)] <- 0

```

	CGcells	CGmedGene	CGmedUMI	BScells	BSmedGene	BSmedUMI	wBScells	wBSmedGene	wBSmedUMI	nBcells	nBSmedGene	nBSmedUMI
<i>i_nb3</i>	172	881	2824	270	426	1012	229	434	1016	41	385	951
<i>L_nc3</i>	89	989	2295	413	372	722	20	450	767	393	366	711
<i>L_nc6</i>	45	998	1868	285	411	729	11	501	921	274	401	722
<i>L_nc5</i>	75	1036	2422	261	447	818	22	500	822	239	439	818
<i>L_nc2</i>	99	1037	2479	448	435	785	23	487	874	425	427	775
<i>L_nc1</i>	59	1063	2463	578	376	706	18	445	778	560	373	698
<i>L_nc7</i>	38	1087	2116	212	404	574	7	532	774	205	403	572
<i>L_nc4</i>	64	1122	2512	293	469	789	14	576	959	279	464	784
<i>L_nc8</i>	42	1146	2500	133	507	968	12	576	1010	121	485	933
<i>i_nem</i>	394	1171	2576	241	520	876	122	499	786	119	592	1042
<i>enEp_SC3</i>	294	1209	2654	178	636	1131	174	640	1146	4	450	684
<i>i_smgo</i>	398	1443	5792	291	730	2074	180	675	1890	111	855	2240
<i>L_zmg2</i>	101	1471	10189	80	810	5768	73	832	5984	7	771	3014
<i>L_nb2</i>	428	1485	3930	248	618	1136	212	622	1138	36	571	1144
<i>L_nb5</i>	245	1531	6362	174	804	2601	162	798	2504	12	910	3228
<i>i_smgo</i>	520	1566	5098	350	756	1867	214	742	1818	136	786	2060
<i>L_nb4</i>	212	1590	4596	108	754	1569	93	770	1550	15	728	1599
<i>db1</i>	130	1700	4301	14	682	1084	14	682	1084	0	0	0
<i>L_mgl</i>	525	1821	4112	14	1382	2875	14	1382	2875	0	0	0
<i>L_zmg1</i>	345	1857	13841	258	834	4870	214	774	4605	44	1366	6514
<i>enEp_tent</i>	282	1902	5522	82	1079	2456	81	1063	2409	1	2065	6304
<i>L_nb1</i>	766	1944	5686	294	798	1488	233	821	1482	61	709	1501
<i>db2</i>	55	2071	6486	1	5258	25465	1	5258	25465	0	0	0
<i>i_nc_prog</i>	566	2244	6351	711	747	1394	98	913	1638	613	717	1359
<i>ecEp_bd</i>	526	2312	7422	118	1134	2628	117	1141	2676	1	431	728
<i>enEp_tent(pd)</i>	110	2394	7216	42	1576	3811	41	1569	3793	1	4352	16904
<i>enEp_SC2</i>	1239	2409	7961	562	1530	3752	560	1530	3752	2	1852	5562
<i>enEp_head</i>	382	2456	6973	98	1272	2709	98	1272	2709	0	0	0
<i>i_gland_prog</i>	81	2553	8267	47	1307	3491	18	1367	3316	29	1288	3592
<i>L_SC</i>	1635	2574	9201	494	1038	2278	311	1009	2107	183	1110	2628
<i>enEp_foot</i>	624	2638	8389	171	1556	3494	169	1556	3494	2	2146	8396
<i>ecEp_SC1</i>	1145	2691	9445	564	1623	4151	562	1630	4172	2	846	1967
<i>enEp_nem1(pd)</i>	237	3158	11420	61	2153	5481	60	2178	5720	1	1940	5289
<i>ecEp_head</i>	648	3296	11932	99	1926	4948	98	1906	4857	1	2279	6486
<i>db3</i>	192	3374	13294	112	2274	6530	101	2310	7225	11	1278	3028
<i>ecEp_battery(mp)</i>	564	3522	12338	105	1967	4596	104	1966	4584	1	4964	22784
<i>L_fmg1</i>	324	3526	11664	218	1760	3790	206	1714	3598	12	2489	7018
<i>ecEp_nem3(mp)</i>	80	3584	11507	2	1623	3291	2	1623	3291	0	0	0
<i>enEp_nem2(pd)</i>	147	3594	14034	35	2138	5798	34	2128	5719	1	2391	6963
<i>ecEp_nem2(mp)</i>	432	3904	15982	107	2376	6558	107	2376	6558	0	0	0
<i>ecEp_SC2</i>	1113	3951	17016	71	2545	8452	71	2545	8452	0	0	0
<i>enEp_SC1</i>	1921	3956	17080	344	2736	8796	336	2730	8706	8	3253	10958
<i>ecEp_nem1(mp)</i>	639	4157	18190	221	2529	8301	219	2529	8301	2	4032	24592
<i>L_fmg2</i>	516	4772	17394	4	2308	4760	4	2308	4760	0	0	0

Figure 11: Median number of genes and UMIs obtained with different beads and cell suspensions. ChemGenes beads, cell suspensions from whole animal dissociation (CG). Biosearch beads (BS), cell suspensions from whole animal dissociation and FACS sorted cells. Biosearch beads (wBS), cell suspensions from whole animal dissociation. Biosearch beads (nBS), neuronal cells enriched using FACS.

```
rownames(wBSnBS) <- wBSnBS$cellState
wBSnBS$cellState <- NULL
```

Cut-off exploration

We next evaluate the impact of cut-off selection for number of genes and UMIs per cell on cells retained in the analysis (Fig. 12). No clear cut-offs are suggested in elbow plots (not shown). Increasing the lower gene cut-off to 400 cells would exclude 1,725 cells, a large fraction of which these are neurons (Fig. 12). C). All identified cell states are present at a lower cut-off of 300 genes per cell. On the high end (> 7000 genes) we find that cells contribute to multiple clusters not suggesting that particular doublet categories can be excluded via cut-off selection. We also explore the cells that each library contributes to the data set (Fig. 13). We find a similar contribution from libraries that were generated using whole animals (Fig. 13 B-G, M-N). In libraries that contained sexually reproducing animals (Fig. 13), H-K) the ratio of germline cells is increased. Neuronal libraries are predominantly composed of neuronal progenitors and differentiated neurons as well as, at lower numbers, of other interstitial somatic cells such as gland cells and nematocytes. For the down-stream analyses we settled on $>300 < 7k$ genes, $>500\text{UMI} < 50\text{k}$ UMIs as cut-offs.

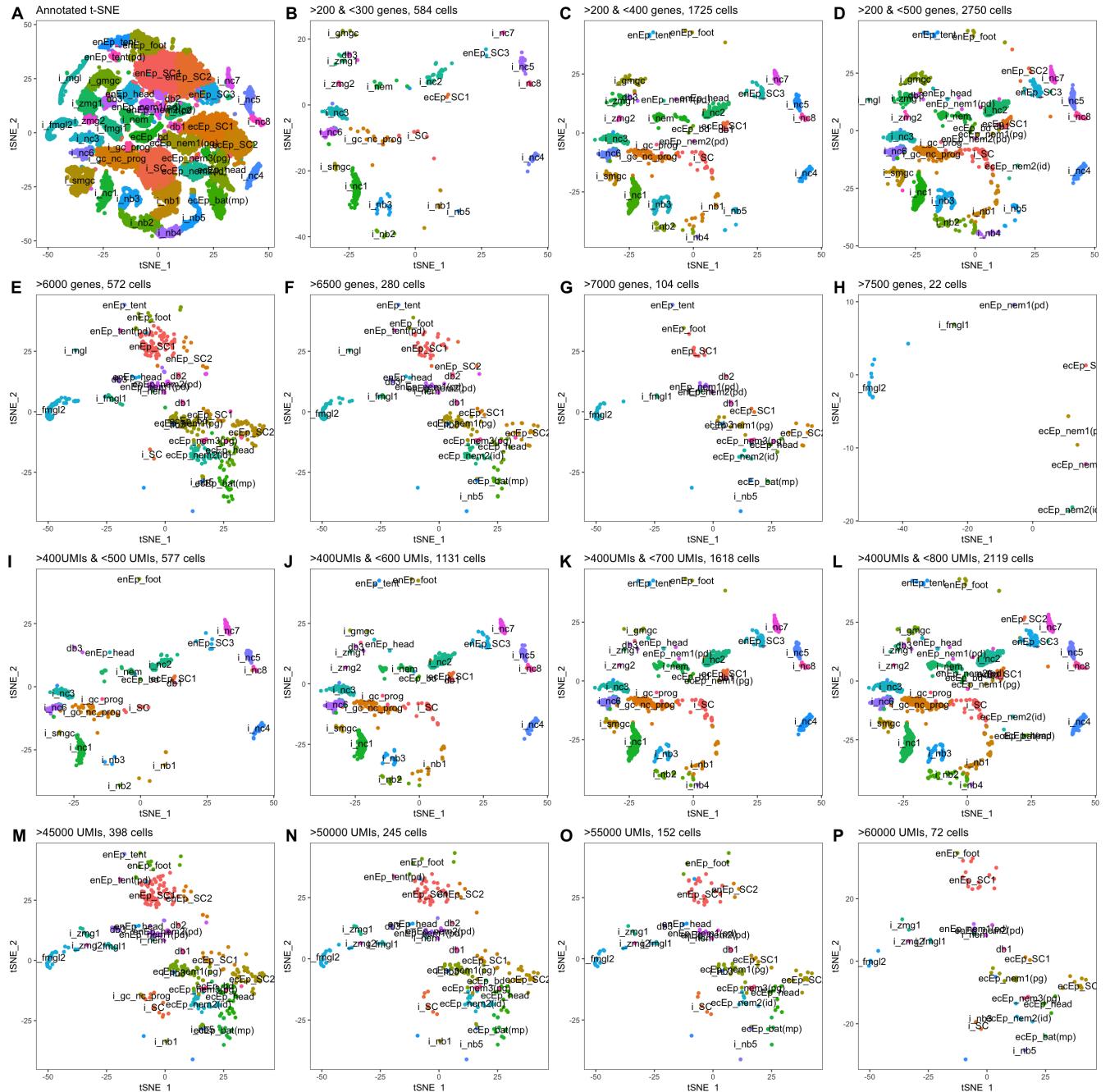


Figure 12: Cut-off exploration. The permissive analysis included cells with > 200 and $< 8k$ genes and $> 400 < 70k$ UMIs. Range of expressed genes/detected UMIs for plotted cells and cell numbers are indicated in titles of individual plots.

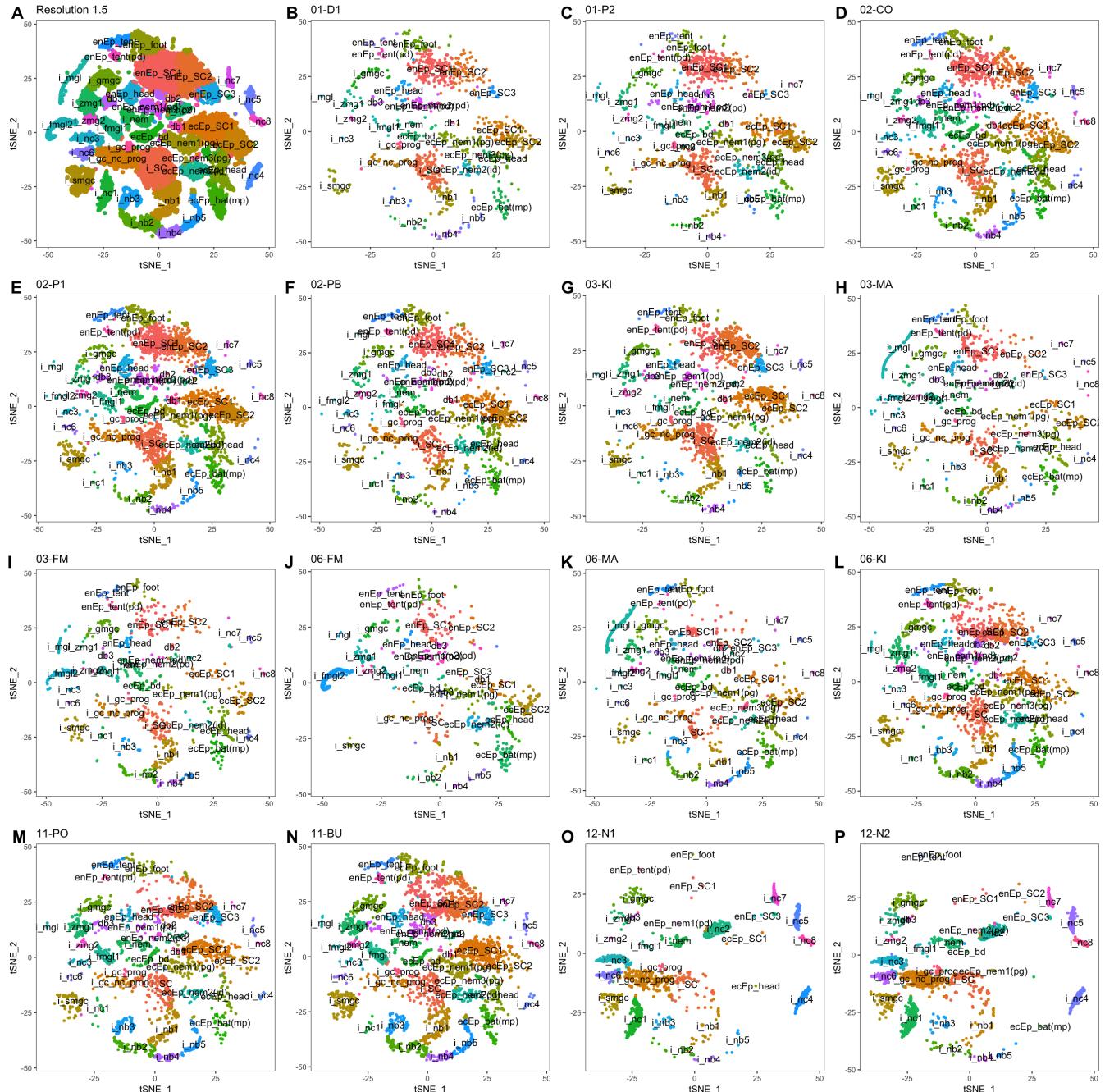


Figure 13: Cells contributed by each library.

0.0.1 Software versions

This document was computed on Thu Nov 01 12:42:39 2018 with the following R package versions.

```
R version 3.4.3 (2017-11-30)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: OS X El Capitan 10.11.6

Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] grid      stats     graphics   grDevices  utils      datasets   methods
[8] base

other attached packages:
[1] bindrcpp_0.2.2 rlang_0.3.0.1  gridExtra_2.3  gtable_0.2.0
[5] dplyr_0.7.7   Seurat_2.2.1   Matrix_1.2-12  cowplot_0.9.2
[9] ggplot2_3.1.0 knitr_1.20

loaded via a namespace (and not attached):
 [1] diffusionMap_1.1-0    Rtsne_0.13          VGAM_1.0-5
 [4] colorspace_1.3-2       ggridges_0.4.1      class_7.3-14
 [7] modeltools_0.2-21     mclust_5.4         rprojroot_1.3-2
[10] htmlTable_1.11.2      base64enc_0.1-3    proxy_0.4-21
[13] rstudioapi_0.7        DRR_0.0.3          flexmix_2.3-14
[16] lubridate_1.7.3       prodlim_1.6.1      mvtnorm_1.0-7
[19] ranger_0.9.0          codetools_0.2-15   splines_3.4.3
[22] R.methodsS3_1.7.1     mnormt_1.5-5       robustbase_0.92-8
[25] tclust_1.3-1          RcppRoll_0.2.2     Formula_1.2-2
[28] caret_6.0-78          ica_1.0-1          broom_0.4.3
[31] ddalpha_1.3.1.1       cluster_2.0.6     kernlab_0.9-25
[34] R.oo_1.21.0           sfsmisc_1.1-2     compiler_3.4.3
[37] backports_1.1.2      assertthat_0.2.0  lazyeval_0.2.1
[40] formatR_1.5            lars_1.2           acepack_1.4.1
[43] htmltools_0.3.6       tools_3.4.3       igraph_1.2.2
[46] glue_1.3.0             reshape2_1.4.3    Rcpp_0.12.19
[49] trimcluster_0.1-2     gdata_2.18.0      ape_5.1
[52] nlme_3.1-131.1        iterators_1.0.9   fpc_2.1-11
[55] psych_1.7.8           timeDate_3043.102 xfun_0.1
[58] gower_0.1.2           stringr_1.3.0      irlba_2.3.2
[61] gtools_3.5.0           DEoptimR_1.0-8     MASS_7.3-49
[64] scales_1.0.0           ipred_0.9-6       parallel_3.4.3
[67] RColorBrewer_1.1-2    yaml_2.1.18       pbapply_1.3-4
[70] segmented_0.5-3.0     rpart_4.1-13      latticeExtra_0.6-28
[73] stringi_1.1.6          highr_0.6         foreach_1.4.4
[76] checkmate_1.8.5        calTools_1.17.1.1 lava_1.6
[79] dtw_1.18-1              SDMTools_1.1-221 pkgconfig_2.0.2
[82] prabclus_2.2-6         bitops_1.0-6       evaluate_0.10.1
[85] lattice_0.20-35        ROCR_1.0-7        purrr_0.2.5
[88] bindr_0.1.1             labeling_0.3      recipes_0.1.2
[91] htmlwidgets_1.0          CVST_0.2-1       tidyselect_0.2.5
[94] plyr_1.8.4              magrittr_1.5      bookdown_0.7
[97] R6_2.3.0                gplots_3.0.1      Hmisc_4.1-1
[100] dimRed_0.1.0            sn_1.5-1         pillar_1.2.1
[103] foreign_0.8-69          withr_2.1.2      mixtools_1.1.0
[106] survival_2.41-3         scatterplot3d_0.3-40 nnet_7.3-12
[109] tsne_0.1-3              tibble_1.4.2      crayon_1.3.4
[112] KernSmooth_2.23-15      rmarkdown_1.9      data.table_1.11.8
[115] FNN_1.1                  ModelMetrics_1.1.0 metap_0.8
[118] digest_0.6.18            diptest_0.75-7    numDeriv_2016.8-1
[121] tidyverse_0.8.0           R.utils_2.6.0      stats4_3.4.3
[124] munsell_0.5.0
```

References

1. R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, A. Regev, Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*. **33**, 495–502 (2015).
2. E. Z. Macosko *et al.*, Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*. **161**, 1202–1214 (2015).
3. R. Augustin *et al.*, Activity of the novel peptide arminin against multiresistant human pathogens shows the considerable potential of phylogenetically ancient organisms as drug sources. *Antimicrobial agents and chemotherapy*. **53**, 5245–5250 (2009).
4. B. Hobmayer *et al.*, WNT signalling molecules act in axis formation in the diploblastic metazoan Hydra. *Nature*. **407**, 186–189 (2000).
5. A. Grens, L. Gee, D. A. Fisher, H. R. Bode, CnNK-2, an NK-2 Homeobox Gene, Has a Role in Patterning the Basal End of the Axis in Hydra. **180**, 473–488 (1996).
6. S. Thomsen, T. C. G. Bosch, Foot differentiation and genomic plasticity in Hydra: lessons from the PPOD gene family. *Development genes and evolution*. **216**, 57–68 (2006).
7. I. Endl, J. U. Lohmann, T. C. Bosch, Head-specific gene expression in Hydra: complexity of DNA-protein interactions at the promoter of ks1 is inversely correlated to the head activation potential. *Proceedings of the National Academy of Sciences of the United States of America*. **96**, 1445–1450 (1999).
8. K. M. Smith, L. Gee, H. R. Bode, HyAlx, an aristaless-related gene, is involved in tentacle formation in hydra. *Development (Cambridge, England)*. **127**, 4743–4752 (2000).
9. K. Mochizuki, H. Sano, S. Kobayashi, C. Nishimiya-Fujisawa, T. Fujisawa, Expression and evolutionary conservation of nanos-related genes in Hydra. *Wilhelm Roux' Archiv für Entwicklungsmechanik der Organismen*. **210**, 591–602 (2000).
10. R. Augustin *et al.*, Dickkopf related genes are components of the positional value gradient in Hydra. **296**, 62–70 (2006).
11. C. Guder *et al.*, An ancient Wnt-Dickkopf antagonism in Hydra. *Development (Cambridge, England)*. **133**, 901–911 (2006).
12. S. Siebert, F. Anton-Erxleben, T. C. G. Bosch, Cell type complexity in the basal metazoan Hydra is maintained by both stem cell based mechanisms and transdifferentiation. *Developmental biology*. **313**, 13–24 (2008).
13. J. S. Hwang *et al.*, Nematogalectin, a nematocyst protein with GlyXY and galectin domains, demonstrates nematocyte-specific alternative splicing in Hydra. *Proceedings of the National Academy of Sciences of the United States of America*. **107**, 18539–18544 (2010).
14. J. S. Hwang *et al.*, Cilium evolution: identification of a novel protein, nematocilin, in the mechanosensory cilium of Hydra nematocytes. *Molecular biology and evolution*. **25**, 2009–2017 (2008).
15. S. Fraune *et al.*, In an early branching metazoan, bacterial colonization of the embryo is controlled by maternal antimicrobial peptides. *Proceedings of the National Academy of Sciences of the United States of America*. **107**, 18067–18072 (2010).
16. R. Campbell, Organization of the nematocyst battery in the tentacle of hydra: Arrangement of the complex anchoring junctions between nematocytes, epithelial cells, and basement membrane. *Cell and Tissue Research*. **249** (1987).
17. Hobmayer, B, Holstein, T W, David, C N, Tentacle morphogenesis in hydra. *Development (Cambridge, England)*. **109**, 897–904 (1990).
18. H. R. Bode, K. M. Flick, Distribution and dynamics of nematocyte populations in *Hydra attenuata*. *Journal of cell science*. **21**, 15–34 (1976).
19. Dübel, Stefan, Cell differentiation in the head of Hydra. *Differentiation*. **41**, 99–109 (1989).
20. M. K. Lyon, G. Kass-Simon, L. A. Hufnagel, Ultrastructural analysis of nematocyte removal in Hydra. *Tissue and Cell*. **14**, 415–424 (1982).
21. Campbell, R D, Elimination of Hydra interstitial and nerve cells by means of colchicine. *Journal of cell science*. **21**, 1–13 (1976).
22. McNeil, P L, Mechanisms of nutritive endocytosis. I. Phagocytic versatility and cellular recognition in Chlorohydra digestive cells, a scanning electron microscope study. *Journal of cell science*. **49**, 311–339

(1981).

23. Seybold, Anna, Salvenmoser, Willi, Hobmayer, Bert, Sequential development of apical-basal and planar polarities in aggregating epitheliomuscular cells of *Hydra*. **412**, 148–159 (2016).