SA06 - Cluster analysis for genome mapped reads

Stefan Siebert

October 25, updated on May 25, 2019

Summary

Drop-seq reads from 15 libraries generated for *Hydra vulgaris* strain AEP were mapped to the 2.0 genome assembly of closely related *Hydra vulgaris* strain 105 (available at https://research.nhgri.nih.gov/hydra/) and processed using the *Hydra* 2.0 gene models. Strain *Hydra vulgaris* 105 was formerly referred to as *Hydra magnipapillata* 105. We performed graph-based clustering of recovered cells using Seurat (1) considering cells with >300 <7k genes and >500UMI < 50k UMIs. NMF analysis was performed to identify modules of co-expressed genes (NMF analysis wg_K84). We apply neuron annotations that were introduced in the neuronal subcluster analysis for transcriptome data (see analysis SA05_SubclustNeuronalCells).

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
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 data

We load `GSE121617_Hydra_DS_genome_UMICounts.txt`, which is a *genes X cell* data.frame of unnormalized, unlogged transcripts detected per gene per cell.

```
# load UMI data
ds.counts <- read.table("objects/GSE121617_Hydra_DS_genome_UMICounts.txt",
    sep = "\t", check.names = FALSE, header = TRUE)
```

Create Seurat object

```
# Keep all genes expressed in >= 3 cells, keep all cells with >= 300
# genes
aep.ds <- CreateSeuratObject(raw.data = ds.counts, min.cells = 3, min.genes = 300,
    project = "Hydra")

ds.ds <- MakeSparse(object = aep.ds)
rm(aep.ds)
```

Clustering of cells

```
# Apply transcriptome gene/UMI cut-offs
ds.ds <- FilterCells(object = ds.ds, subset.names = c("nGene", "nUMI"),
    low.thresholds = c(300, 500), high.thresholds = c(7000, 50000))
# Normalize the data
ds.ds <- NormalizeData(object = ds.ds, normalization.method = "LogNormalize",
    scale.factor = 10000)
```

```
# Find 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)
# Run 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, genes.print = 20)
# Project PCA to find genes that weren't scored as highly variable, but
# should belong to a given PC and include them.
ds.ds <- ProjectPCA(object = ds.ds)

# perform permutation test to directly calculate p-value ds.ds <-
# JackStraw(object = ds.ds, num.pc = 40, num.replicate = 100, do.print
# = FALSE) JackStrawPlot(object = ds.ds, PCs=1:40)

# Approximate amount of variance encoded by each PC
PCElbowPlot(object = ds.ds, num.pc = 40)

# Find cluster
ds.ds <- FindClusters(object = ds.ds, reduction.type = "pca", dims.use = 1:30,
    force.recalc = TRUE, resolution = 1.5, print.output = 0)
# Run TSNE
ds.ds <- RunTSNE(object = ds.ds, dims.use = c(1:30), do.fast = T)

# saveRDS('objects/genome_pc30.rds')
```

Doublet identification

Analogous to filtering performed in case of transcriptome mapped data, we exclude cells that express male germ cell specific histones and that are not part of the male germline cluster (see SA02_ClustTranscriptome). We also exclude suspected doublet cells that co-express endodermal and zymogen gland cell markers as well as cells that co-express ectodermal and ectodermal epithelial cell markers.

```
# We load the object from our original analysis
ds.ds <- readRDS("objects/genome_pc30.rds")

# Exclude histone positive cells outside the male germline cluster

# Isolate male germline cluster, cluster 16 in our original analysis
ds.male <- SubsetData(object = ds.ds, ident.use = c(16), subset.raw = TRUE)

## There are a few histone positive cells that are assigned to the
## germline clusters but group with epithelial cells.  We manually
## select these cells using the argument do.identify.  select.cells <-
## TSNEPlot(object = ds.male, do.identify = TRUE)

# Load cell ids that were excluded in our original analysis
select.cells <- c("02-P1_TATTTGCCGTTC", "03-MA_CCATGTGACTCG", "03-MA_TAACTTATGAGA",
    "06-MA_TCGGACTTTCTT", "06-MA_AGTTCCTTCCGA", "06-MA_ATCCCTAGTGAC", "06-MA_CTCAATCCCGGT",
    "06-MA_ACACCCTGTCGA", "06-MA_ACAGCGGCCCTC", "06-MA_ATGCCTATGTCC", "06-MA_CTCATCCCGGTN",
    "06-MA_GTGGATGCAAAT", "06-MA_CTAGGGTCTACN", "06-MA_CGCGGGGTACAC", "06-MA_GGTGGTGAGAGN",
    "06-MA_CTAAGGGTCTAC", "06-MA_CGCGCTCACGGA", "06-MA_GCGCCAAACGCG", "06-MA_CGTTTAACCGTA",
    "06-MA_TTTCTCCATTGC", "06-MA_TCGCTGCCCACA", "06-MA_AGGTATCAGATT", "06-MA_AACTATTGACAC",
    "06-MA_ACTGTCATCGTT", "06-MA_CAACACCTATAC", "06-MA_TAGTTTACTGCC", "06-MA_CTCTACGTCAAT",
    "11-PO_CATCAGGAGATT", "11-BU_ATCCACTGTCCA")

# IDs for all cells from male germline cluster
cells <- ds.male@data@Dimnames[[2]]
# Identify cells to keep
cells.keep <- setdiff(cells, select.cells)
# Update subset of cells from the male germline
ds.male <- SubsetData(object = ds.male, cells.use = cells.keep, subset.raw = TRUE)

# Combine ids for suspected doublets
db <- c(gland.endo, endo.ecto)

# All cell IDs
cells <- ds.ds@data@Dimnames[[2]]
# Identify cells to keep
```

```r
cells.keep <- setdiff(cells, db)

# Remove doublets
ds.cl <- SubsetData(object = ds.ds, cells.use = cells.keep, subset.raw = TRUE)

# Get all cells not part of the male germline cluster, cluster 16 in
# our original analysis
ds.else <- SubsetData(object = ds.cl, ident.remove = c(16), subset.raw = TRUE)

# Identify cells with H2BE expression < 0.5
gate1 <- WhichCells(ds.else, subset.name = "g1835.t1|H2BE_STRPU", accept.high = 0.5)

# Identify cells with H10/H5 expression < 0.5
gate2 <- WhichCells(ds.else, subset.name = "g32541.t1|H5_CHICK", accept.high = 0.5)

# Identify cells not expressing either H2BE or H10A/H5 above the cutoff
cells.keep.else <- intersect(gate1, gate2)

# Filtered subset, doublets, germline cluster expcluded
ds.else <- SubsetData(ds.else, cells.use = cells.keep.else, subset.raw = TRUE)
# Add germline cluster cells
ds.g.s1 <- MergeSeurat(ds.male, ds.else)

# Remove objects
rm(ds.ds)
rm(ds.male)
rm(ds.else)
```

Clustering

At this stage the data set includes 24458 cells with a median of 1273 genes and a median of 3488 UMIs per cell. We cluster the cells as before, annotate the t-SNE representation (Figs. 1, 2) and plot metagenes from NMF analysis wg_K84 (Fig. 3).

```r
# We cluster the remaining cells

# Find variable genes
ds.g.s1 <- FindVariableGenes(object = ds.g.s1, mean.function = ExpMean,
    dispersion.function = LogVMR, x.low.cutoff = 0.05, x.high.cutoff = 4,
    y.cutoff = 0.5)
# Scale
ds.g.s1 <- ScaleData(object = ds.g.s1)
# Do PCA on highly variable genes
ds.g.s1 <- RunPCA(object = ds.g.s1, pc.genes = ds.g.s1@var.genes, pcs.compute = 40,
    do.print = TRUE, pcs.print = 1:5, genes.print = 20)
# Project PCA to find genes that weren't scored as highly variable, but
# should belong to a given PC and include them.
ds.g.s1 <- ProjectPCA(object = ds.g.s1)

# Perform permutation test to directly calculate p-value ds.g.s1 <-
# JackStraw(object = ds.g.s1, num.pc = 40, num.replicate = 100,
# do.print = FALSE) JackStrawPlot(object = ds.g.s1, PCs=1:40)

# Approximation of amount of variance encoded by each PC
PCElbowPlot(object = ds.g.s1, num.pc = 40)

# Find cluster
ds.g.s1 <- FindClusters(object = ds.g.s1, reduction.type = "pca", dims.use = 1:30,
    force.recalc = TRUE, resolution = 1.5)
ds.g.s1 <- RunTSNE(object = ds.g.s1, dims.use = c(1:30), do.fast = T, perplexity = 40)

TSNEPlot(object = ds.g.s1, group.by = "res.1.5", do.return = T, do.label = T)

# saveRDS(ds.ds,'objects/genome_S1_pc30.rds')
```

```r
# CLUSTER ANNOTATION

# Multiple cluster annotations were used in the course of this study.
# We store them in the Seurat object.  They can be restored using
# Seurat function SetAllIdent(), e.g. ds.g.s1 <- SetAllIdent(object =
# ds.g.s1, id = 'cluster.manuscript')

# We load the object from our original analysis
ds.g.s1 <- readRDS("objects/genome_S1_pc30.rds")

# Store cluster numbering
ds.g.s1 <- StashIdent(object = ds.g.s1, save.name = "cluster_numbering")

current.cluster.ids <- as.character(0:40)

# Run this to restore original cluster numbering
ds.g.s1 <- SetAllIdent(object = ds.g.s1, id = "cluster_numbering")

# Cluster labeling scheme used in the bioRxiv preprint (Fig. S24)
cluster.names <- c("ecEp_SC1", "enEp_SC1", "enEp_SC2", "i_SC", "i_nb1",
    "ecEp_nem1(pd)", "i_nc_prog", "enEp_SC3", "i_smgc", "ecEp_bat(mp)",
    "ecEp_head/hyp", "enEp_head", "i_nb2", "i_nb4", "enEp_tent", "i_fmgl1",
    "i_fmgl2", "i_gmgc", "i_mgl", "i_zmg2", "i_nb3", "ecEp_nem2(id)", "enEp_foot",
    "ecEp_bd", "i_nem", "i_gc_nc_prog", "i_nc2", "ecEp_SC2", "enEp_nem(pd)",
    "i_nc1", "i_nc8", "i_nc6", "ecEp_ped", "i_zmg1", "i_nc7", "i_nb5",
    "i_nc3", "i_nc4", "unident", "i_nc5", "unident2")

# update names in Seurat object
ds.g.s1@ident <- plyr::mapvalues(x = ds.g.s1@ident, from = current.cluster.ids,
    to = cluster.names)

# Stash labels
ds.g.s1 <- StashIdent(object = ds.g.s1, save.name = "cluster.preprint")

# Match genome neuron clusters to neuron clusters recovered in the
# transriptome analysis

# nc1 - n_ec3, nc2 - n_ec1, nc3 - n_ec4, nc4 - n_ec5, nc5 - n_en3, nc6
# - n_en1, nc7 - n_en2, nc8 - n_ec2.

# Run this to restore original cluster numbering
ds.g.s1 <- SetAllIdent(object = ds.g.s1, id = "cluster_numbering")

# Modified cluster labeling with consistent neuron labels across
# clusterings, short labels (useful when working in R).  Cluster 38 is
# positive for ectodermal epithelial genes but but also for genes known
# to be expressed in female germline cells, e.g. periculin.  We
# tentatively label it here as ecEp-fmgl(db). Analogous to uptake of
# nematoblast epithelial cells may phagocytose nurse cells.
cluster.names <- c("ecEp_SC1", "enEp_SC1", "enEp_SC2", "i_SC", "i_nb1",
    "ecEp-nb(pd)", "i_n_prog", "enEp_SC3", "i_smgc", "ecEp_bat(mp)", "ecEp_head/hyp",
    "enEp_head", "i_nb2", "i_nb4", "enEp_tent", "i_fmgl1", "i_fmgl2", "i_gmgc",
    "i_mgl", "i_zmg1", "i_nb3", "ecEp-nem(id)", "enEp_foot", "ecEp_bd",
    "i_nem", "i_gc/n_prog", "i_n_ec1", "ecEp_SC2", "enEp-nem(pd)", "i_n_ec3",
    "i_n_ec2", "i_n_en1", "ecEp_ped", "i_zmg2", "i_n_en2", "i_nb5", "i_n_ec4",
    "i_n_ec5", "ecEp-fmgl(db)", "i_n_en3", "unident")

# update names in Seurat object
ds.g.s1@ident <- plyr::mapvalues(x = ds.g.s1@ident, from = current.cluster.ids,
    to = cluster.names)

TSNEPlot(object = ds.g.s1, do.return = T, do.label = T, no.legend = TRUE,
    return = FALSE, label.size = 5.5, pt.size = 0.5)
```

```
# Stash labels
ds.g.s1 <- StashIdent(object = ds.g.s1, save.name = "cluster.short")

# Run this to restore original cluster numbering
ds.g.s1 <- SetAllIdent(object = ds.g.s1, id = "cluster_numbering")

# Modified cluster labeling with consistent neuron labels across
# clusterings and long labels for Broad portal presentation
cluster.names <- c("ecEp_stem_cell_1", "enEp_stem_cell_1", "enEp_stem_cell_2",
    "i_stem_cell/progenitor", "i_nematoblast_1", "ecEp-nb(pd)", "i_neuron_prog",
    "enEp_stem_cell_3", "i_spumous_mucous_gland_cell", "ecEp_battery(mp)",
    "ecEp_head/hypostome", "enEp_head", "i_nematoblast_2", "i_nematoblast_4",
```

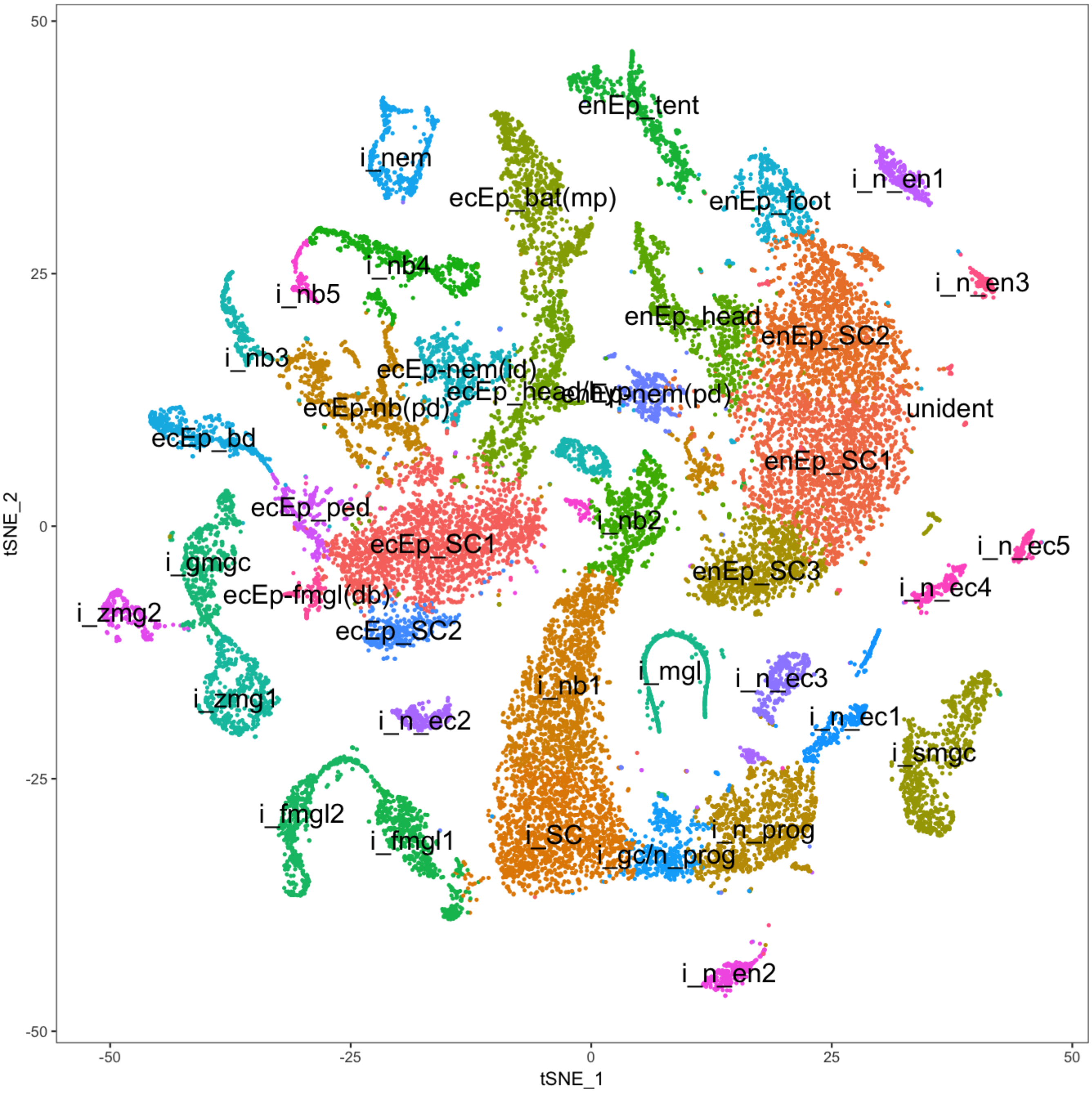

Figure 1: Annotated t-SNE plot. bat: battery cell, db: doublet, ec: ectodermal, en: endodermal, Ep: epithelial cell, fmgl: female germline, gc: gland cell, gmgc: granular mucous gland cell, i: cells of the interstitial lineage, id: integration doublet, mgl: male germline, mp: multiplet, nb: nematoblast, n: neuron, nem: nematocyte, pd: suspected phagocytosis doublet, prog: progenitor, SC: stem cell, smgc: spumous mucous gland cell, tent: tentacle, unident: unidentified, zmg: zymogen gland cell.

```
    "enEp_tentacle", "i_female_germline_1", "i_femalge_germline_2_nurse_cell",
    "i_granular_mucous_gland_cell", "i_male_germline", "i_zymogen_gland_cell_1",
    "i_nematoblast_3", "ecEp-nem(id)", "enEp_foot", "ecEp_basal_disk",
    "i_nematocyte", "i_gland_cell/neuron_progenitor", "i_neuron_ec1", "ecEp_stem_cell_2",
    "enEp-nem(pd)", "i_neuron_ec3", "i_neuron_ec2", "i_neuron_en1", "ecEp_peduncle",
    "i_zymogen_gland_cell_2", "i_neuron_en2", "i_nematoblast_5", "i_neuron_ec4",
    "i_neuron_ec5", "ecEp-fmgl(db)", "i_neuron_en3", "unident")

# update names in Seurat object
ds.g.s1@ident <- plyr::mapvalues(x = ds.g.s1@ident, from = current.cluster.ids,
    to = cluster.names)

# Stash labels
ds.g.s1 <- StashIdent(object = ds.g.s1, save.name = "cluster.long.portal")

# Run this to restore original cluster numbering
ds.g.s1 <- SetAllIdent(object = ds.g.s1, id = "cluster_numbering")

# Modified cluster labeling used in manuscript (Fig. S31B).

cluster.names <- c("ecEp_stem_cell", "enEp_stem_cell", "enEp_stem_cell",
    "i_stem_cell/progenitor", "i_nb1", "ecEp-nb(pd)", "i_neuron_prog",
    "enEp_stem_cell", "i_spumous_mucous_gland_cell", "ecEp_battery(mp)",
    "ecEp_head/hypostome", "enEp_head", "i_nb2", "i_nb4", "enEp_tentacle",
    "i_female_germline_1", "i_femalge_germline_2_nurse_cell", "i_granular_mucous_gland_cell",
    "i_male_germline", "i_zymogen_gland_cell", "i_nb3", "ecEp-nem(id)",
    "enEp_foot", "ecEp_basal_disk", "i_nematocyte", "i_gland_cell/neuron_progenitor",
    "i_neuron_ec1", "ecEp_stem_cell", "enEp-nem(pd)", "i_neuron_ec3", "i_neuron_ec2",
    "i_neuron_en1", "ecEp_peduncle", "i_zymogen_gland_cell", "i_neuron_en2",
    "i_nb5", "i_neuron_ec4", "i_neuron_ec5", "ecEp-fmgl(db)", "i_neuron_en3",
    "unident")

# update names in Seurat object
ds.g.s1@ident <- plyr::mapvalues(x = ds.g.s1@ident, from = current.cluster.ids,
    to = cluster.names)

# Stash labels
ds.g.s1 <- StashIdent(object = ds.g.s1, save.name = "cluster.manuscript")
```

```
# Load metagene scores for each cell
cellScores <- read.csv("nmf/wg_K84/GoodMeta_CellScores.csv", row.names = 1,
    check.names = F)
cellScores <- as.data.frame(cellScores)
# Make metagenes columns
cellScores <- t(cellScores)
# Fix cell ids
rownames(cellScores) <- sub("X", "", rownames(cellScores))
rownames(cellScores) <- sub("\\.", "-", rownames(cellScores))

# Add scores
cellScores.ds.g.s1 <- cellScores[match(rownames(ds.g.s1@meta.data), rownames(cellScores)),
    ]
ds.g.s1@meta.data <- cbind(ds.g.s1@meta.data, cellScores.ds.g.s1)


# Visualize expression of selected metagenes/gene modules.
p1 <- TSNEPlot(object = ds.g.s1, do.label = T, label.size = 3, pt.size = 0.5,
    cex.names = 6, no.legend = TRUE, do.return = TRUE)
p2 <- FeaturePlot(ds.g.s1, c("wg45", "wg64", "wg17", "wg31", "wg5", "wg47",
    "wg35", "wg74", "wg42", "wg32", "wg13", "wg27", "wg52", "wg56", "wg12",
    "wg76", "wg49", "wg61", "wg15", "wg38", "wg18", "wg59", "wg24", "wg26"),
    cols.use = c("grey", "blue"), do.return = TRUE)

# generate plotlist
plotlist <- prepend(p2, list(p1))

plot_grid(plotlist = plotlist, labels = "AUTO", label_size = 25, align = "h",
    ncol = 5)
```

```
# saveRDS(ds.ds,'objects/Hydra_Seurat_Whole_Genome.rds'))
```

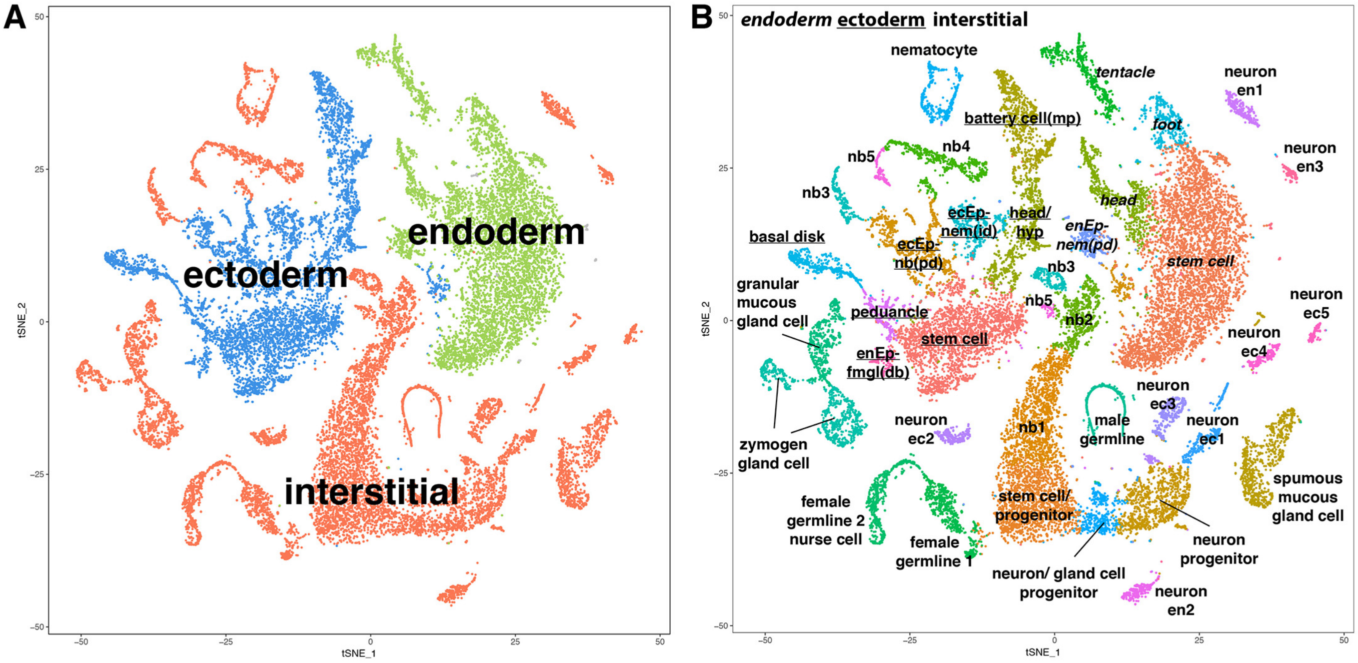

Figure 2: Annotated t-SNE plot with labels edited for clarity (manuscript Fig. S31). A) t-SNE representation of clustered cells colored by cell lineage. B) t-SNE representation of clustered cells annotated with cell states. db: doublet, ec: ectodermal, en: endodermal, Ep: epithelial cell, gc: gland cell, hyp: hypostome, id: integration doublet, mp: multiplet, nb: nematoblast, nem: nematocyte, pd: suspected phagocytosis doublet, prog: progenitor. id, mp, and pd are categories of biological doublets.

Software versions

This document was computed on Sun May 26 15:35:06 2019 with the following R package versions.

```
R version 3.4.4 (2018-03-15)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS  10.14.2

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_2.3.4   Matrix_1.2-15  cowplot_0.9.2
 [9] ggplot2_3.1.0  knitr_1.20

loaded via a namespace (and not attached):
  [1] Rtsne_0.13          colorspace_1.3-2    class_7.3-14
  [4] modeltools_0.2-21   ggridges_0.4.1      mclust_5.4
  [7] rprojroot_1.3-2     htmlTable_1.11.2    base64enc_0.1-3
 [10] rstudioapi_0.7      proxy_0.4-21        npsurv_0.4-0
 [13] flexmix_2.3-14      bit64_0.9-7         mvtnorm_1.0-7
 [16] codetools_0.2-15    splines_3.4.4       R.methodsS3_1.7.1
 [19] lsei_1.2-0          robustbase_0.92-8   Formula_1.2-2
 [22] jsonlite_1.5        ica_1.0-1           cluster_2.0.6
 [25] kernlab_0.9-25      png_0.1-7           R.oo_1.21.0
```

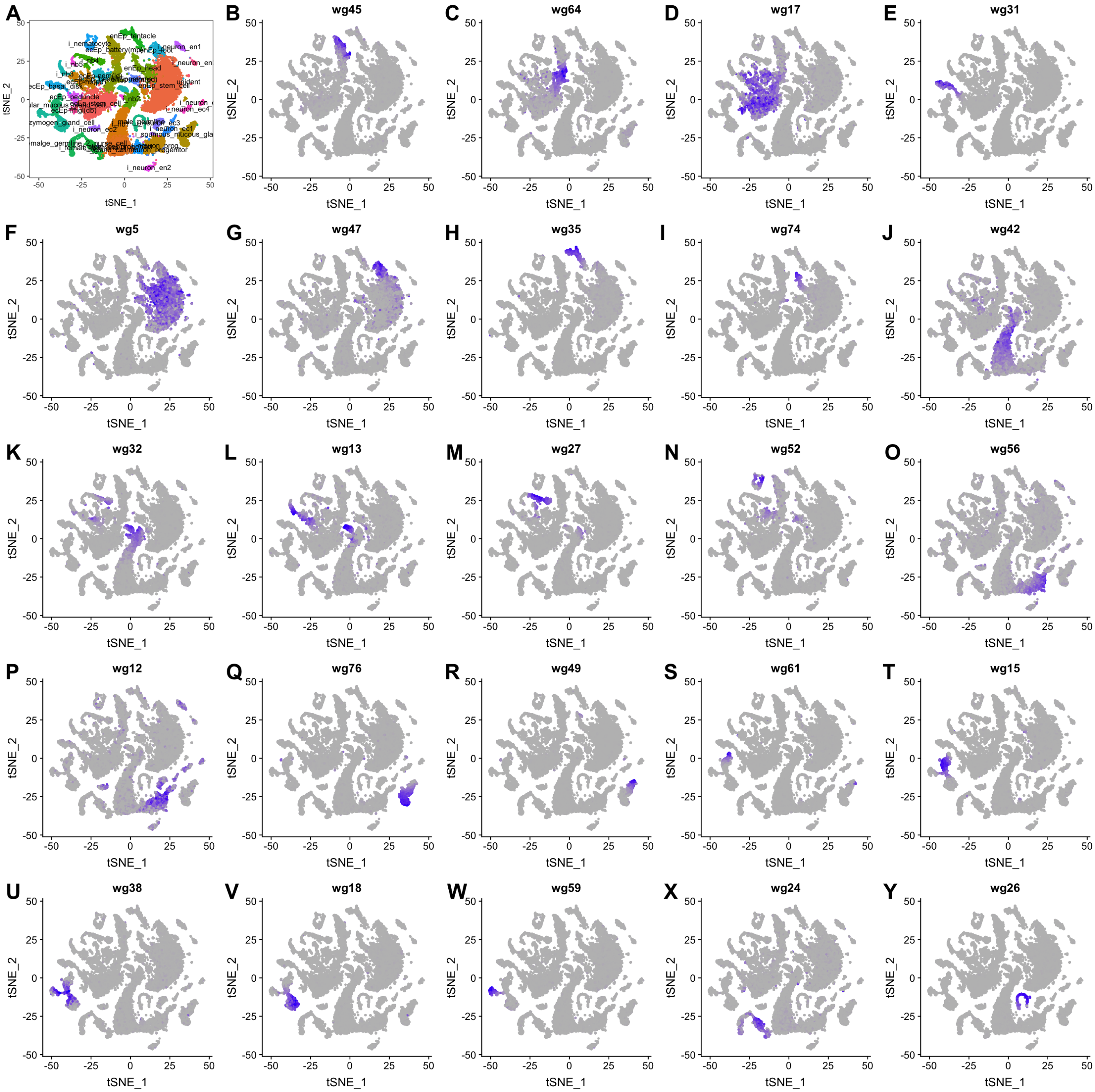

Figure 3: Selected metagenes identified in NMF analysis for the whole dataset after mapping to the Hydra 2.0 genome (wg K84). A metagene describes a set of genes that are co-expressed in the highlighted cell population. A) Annotated t-SNE plot. B) Tentacle ectodermal epithelial cells. This metagene includes transcripts that are expressed in the epithelial cell of a battery cell complex since expression is not found in neuronal or nematocyte cell populations. C) Ectodermal epithelial cells, head/hypostome. D) Ectodermal epithelial cells, body column. E) Ectodermal epithelial cells, basal disk. F) Endodermal epithelial cells, body column. G) Endodermal epithelial cells, foot. H) Endodermal epithelial cells, tentacle. I) Endodermal epithelial cells, hypostome. J) Early stage nematoblast, free and phagocytosed. K) Mid stage nematoblast, free and phagocytosed. L,M) Late nematoblast, singletons and integrated. N) Mature nematocyte, singletons and integrated. O) Neuronal cell progenitors. P) Differentiated neurons, progenitors. Q,R) Spumous mucous gland cells. S) Granular mucous gland cells, hypostome. T) Granular mucous gland cells, mid/lower head. U) Granular mucous gland cells/zymogen gland cells. V,W) Zymogen gland cells. X) Female germline cells. Y) Male germline cells.

```
 [28] compiler_3.4.4    httr_1.3.1         backports_1.1.2
 [31] assertthat_0.2.0  lazyeval_0.2.1     formatR_1.5
 [34] lars_1.2          acepack_1.4.1      htmltools_0.3.6
 [37] tools_3.4.4       igraph_1.2.2       glue_1.3.0
 [40] RANN_2.6          reshape2_1.4.3     Rcpp_0.12.19
 [43] trimcluster_0.1-2 gdata_2.18.0       ape_5.1
 [46] nlme_3.1-131.1    iterators_1.0.9    fpc_2.1-11
 [49] lmtest_0.9-35     xfun_0.1           stringr_1.3.0
 [52] irlba_2.3.2       gtools_3.5.0       DEoptimR_1.0-8
 [55] MASS_7.3-49       zoo_1.8-1          scales_1.0.0
 [58] doSNOW_1.0.16     parallel_3.4.4     RColorBrewer_1.1-2
 [61] yaml_2.1.18       reticulate_1.10    pbapply_1.3-4
 [64] rpart_4.1-13      segmented_0.5-3.0  latticeExtra_0.6-28
 [67] stringi_1.1.6     highr_0.6          foreach_1.4.4
 [70] checkmate_1.8.5   caTools_1.17.1.1   SDMTools_1.1-221
 [73] pkgconfig_2.0.2   dtw_1.18-1         prabclus_2.2-6
 [76] bitops_1.0-6      evaluate_0.10.1    lattice_0.20-35
 [79] ROCR_1.0-7        purrr_0.2.5        bindr_0.1.1
 [82] labeling_0.3      htmlwidgets_1.0    bit_1.1-12
 [85] tidyselect_0.2.5  plyr_1.8.4         magrittr_1.5
 [88] bookdown_0.7      R6_2.3.0           snow_0.4-2
 [91] gplots_3.0.1      Hmisc_4.1-1        pillar_1.2.1
 [94] foreign_0.8-69    withr_2.1.2        fitdistrplus_1.0-11
 [97] mixtools_1.1.0    survival_2.41-3    nnet_7.3-12
[100] tibble_1.4.2      tsne_0.1-3         crayon_1.3.4
[103] hdf5r_1.0.1       KernSmooth_2.23-15 rmarkdown_1.9
[106] data.table_1.11.8 metap_0.8          digest_0.6.18
[109] diptest_0.75-7    tidyr_0.8.0        R.utils_2.6.0
[112] stats4_3.4.4      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).