SA06 - Cluster analysis for genome mapped reads

Stefan Siebert

October 25, 2018

Summary

Drop-seq reads from 15 libraries generated for *Hydra vulgaris* AEP were mapped to the *Hydra vulgaris 105* 2.0 Genome assembly (available at https://research.nhgri.nih.gov/hydra/) and processed using the *Hydra* 2.0 gene models. This genome assembly is for a strain which is closely related to *Hydra vulgaris* AEP and 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).

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

```
# Use 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 <- PCA(ds.ds, do.print = TRUE,
# pcs.print = 20, genes.print = 20)
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 for transcriptome data, we exclude cells that express male germ cell specific histones and that are not part of the male germline cluster (see SA02_ClustTranscriptome.Rmd). 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")

# Excluding 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 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")

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

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

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

```
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)

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 (Fig. 1) and plot metagenes from NMF analysis wg_K84 (Fig. 2).

```
# 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')
```

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

# LABELING CLUSTER, store cluster numbering
ds.g.s1 <- StashIdent(object = ds.g.s1, save.name = "cluster_numbering")
ds.g.s1 <- SetAllIdent(ds.g.s1, "res.1.5")
```

```
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.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)

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

```
# 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)

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'))
```

Software versions

This document was computed on Thu Nov 01 12:20:55 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:
```

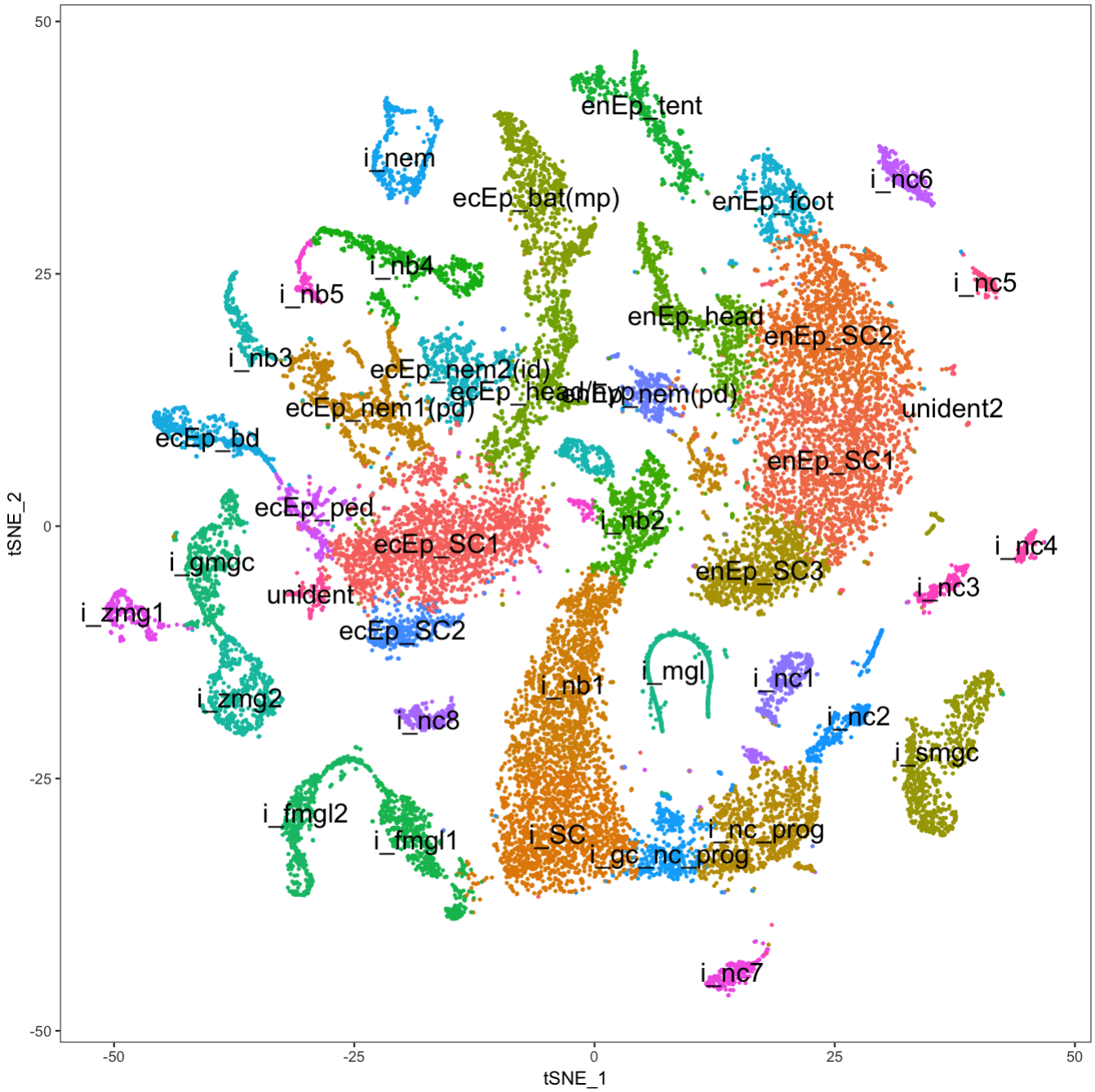

Figure 1: Annotated t-SNE plot. bat: battery cell, db: doublet, ecEP: ectodermal epithelial cell, enEP: endodermal epithelial cell, fmgl: female germ-line, gc: gland cell, gmgc: granular, i: cells of the interstitial lineage, id: integration doublet, mucous gland cell, mgl: male germline, mp: multiplet,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

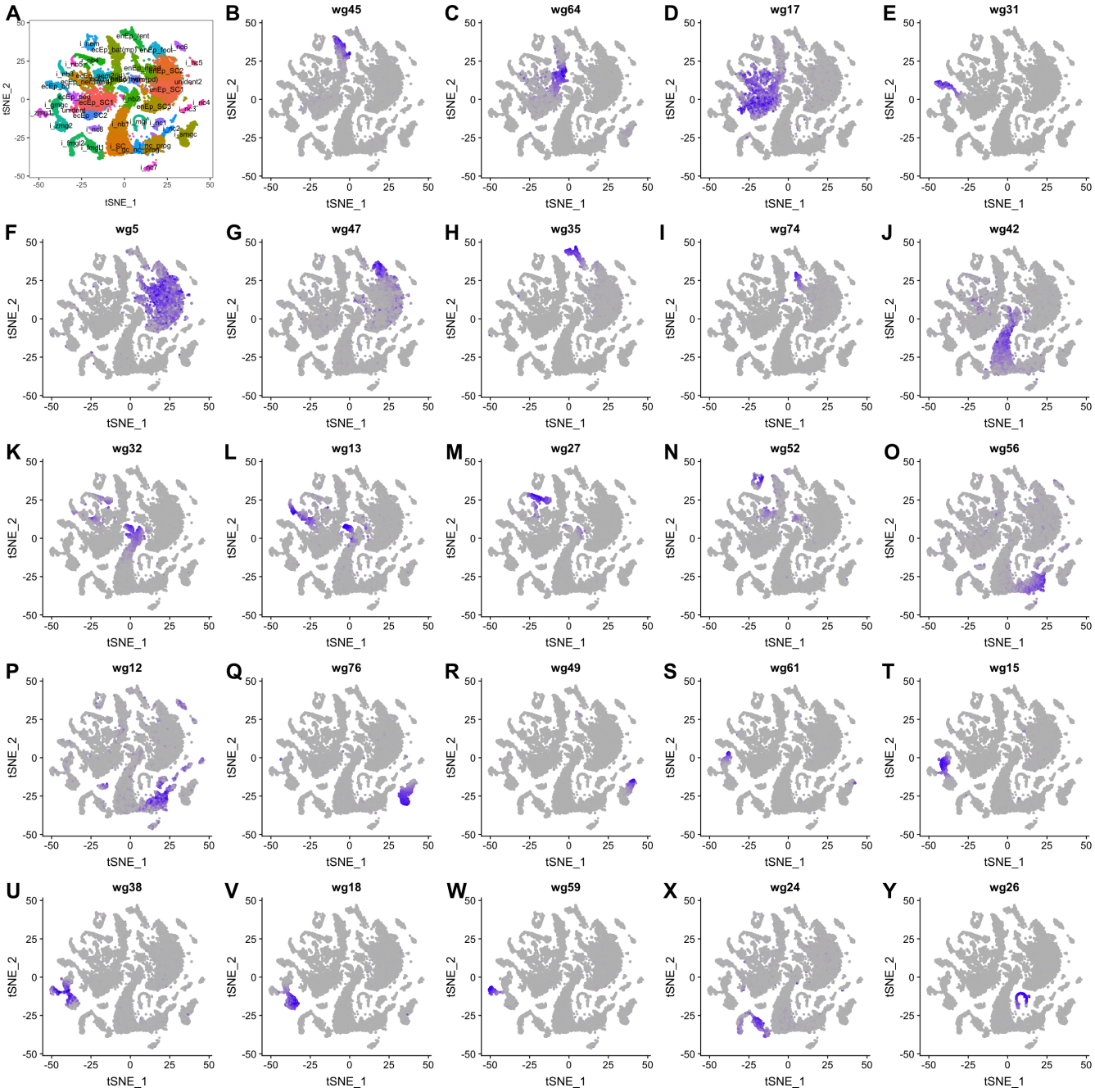

Figure 2: Selected metagenes identified in NMF analysis for the whole dataset after mapping to the Hydra 2.0 genome (wg K84). A metagene describes a set of genes that are co-expressed in the highlighted cell population. A) Annotated t-SNE plot. B) Tentacle ectodermal epithelial cells. This metagene includes 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.

```
 [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      caTools_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] tidyr_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).