SA05 - Subclustering of neuronal cells

Stefan Siebert

October 25, 2018

Summary

We subcluster cells from the neuronal cell lineage. Fluorescent activated cell sorting (FACS) as an additional step in the workflow introduces library specific (batch) effects in case of neuronal libraries 12- which we address in this subclustering. We are able to integrate cells from FACS libraries neuronal cells from whole animal libraries using canonical correlation analysis (CCA) (*1*).

Preliminaries

```
library(edgeR)
library(agalmar)
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.ic@data@Dimnames[[1]][grep(x, ds.ic@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.  We also need epithelial expression
# estimates: GSE121617_DGE_ecto_endo_aepLRv2.txt
```

Subsetting - neuronal cells

We load the interstitial data set and extract neuronal clusters (Fig. 1). We include progenitor cells from clusters 2 and 8.

```
# Load insterstitial object
ds.ic <- readRDS("objects/Hydra_Seurat_IC.rds")

# Choose resolution, in case multiple resolutions were run
ds.ic <- SetAllIdent(ds.ic, "res.1.5")

# Neuronal cell clusters
ds.nc <- SubsetData(object = ds.ic, ident.use = c("2", "8", "15", "24", "11", "14",
    "25", "18", "27", "20", "21"), subset.raw = TRUE)

# > length(ds.nc@meta.data$nGene) [1] 3726

p1 <- TSNEPlot(object = ds.ic, group.by = "res.1.5", do.return = T, do.label = T,
    no.legend = TRUE)
p2 <- TSNEPlot(object = ds.nc, group.by = "res.1.5", do.return = T, do.label = T,
    no.legend = TRUE)

plot_grid(p1, p2, ncol = 2, labels = "AUTO", label_size = 20, align = "h")
```

Clustering of cells without batch regression

We find that FACS enriched neuronal cells separate from neuronal cells that were not sorted prior to performing Drop-seq (Fig. 2).

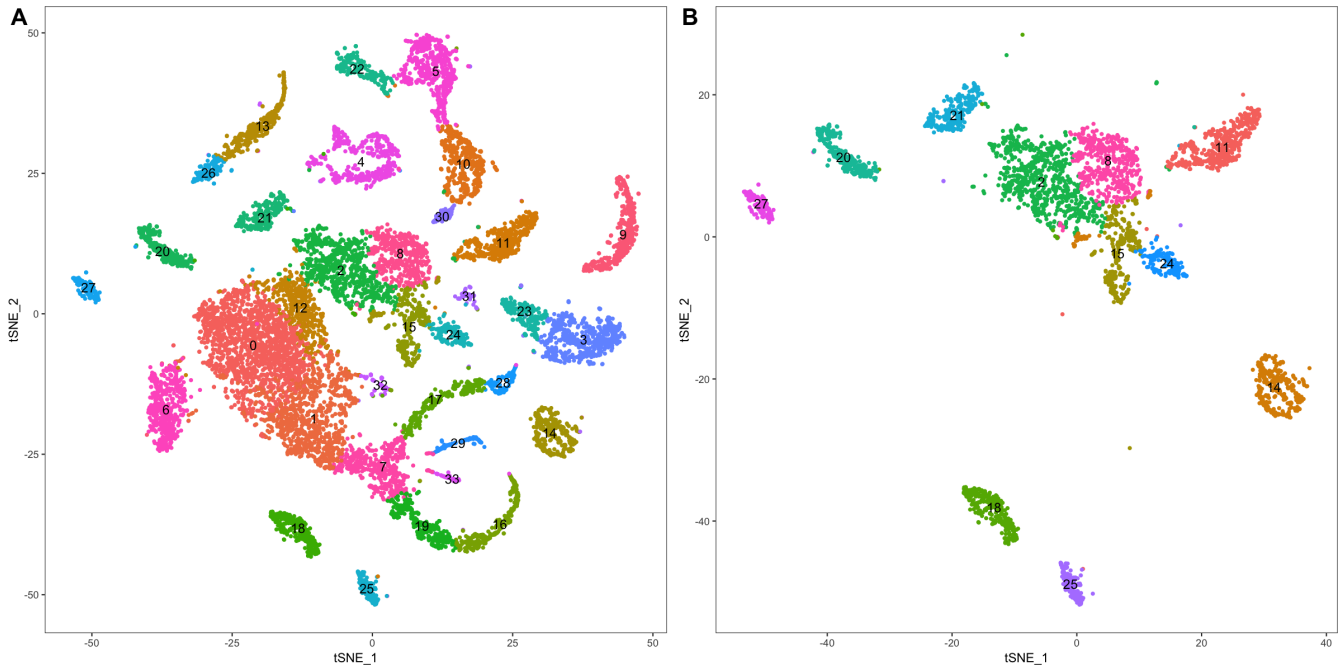

Figure 1: Subsetting cells of the neuronal lineage. A) t-SNE plot for cells of the interstitial lineage, B) Neuronal clusters including differentiated cells and progenitors.

```
# We cluster the cells without batch regression Identify variable genes
ds.nc <- FindVariableGenes(object = ds.nc, mean.function = ExpMean, dispersion.function = LogVMR,
    x.low.cutoff = 0.05, x.high.cutoff = 4, y.cutoff = 0.5)
# 5196 genes identified. More restrictive cut-offs and considering a lower number
# of variable genes yield similar results.  x.low.cutoff = 0.15, x.high.cutoff =
# 4, y.cutoff = 0.5, 2351 variable genes.

# Scale
ds.nc <- ScaleData(object = ds.nc)
# PCA on highly variable genes
ds.nc <- RunPCA(object = ds.nc, pc.genes = ds.nc@var.genes, pcs.compute = 40, do.print = TRUE,
    pcs.print = 1:5, genes.print = 20)
# Project PCA
ds.nc <- ProjectPCA(object = ds.nc)

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

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

ds.nc <- FindClusters(object = ds.nc, reduction.type = "pca", dims.use = 1:22, force.recalc = TRUE,
    resolution = 1.5, print.output = 0)
ds.nc <- RunTSNE(object = ds.nc, dims.use = c(1:22), do.fast = T)

# saveRDS(ds.nc,'objects/ds.nc.pc22.rds')
```

```
# Load object from original analysis
ds.nc <- readRDS("objects/ds.nc.pc22.rds")

p1 <- TSNEPlot(object = ds.nc, do.return = T, do.label = T, no.legend = TRUE)
p2 <- TSNEPlot(object = ds.nc, do.return = T, group.by = "orig.ident")
p3 <- FeaturePlot(object = ds.nc, do.return = T, features.plot = "nUMI", cols.use = c("grey",
    "red"))
p4 <- FeaturePlot(object = ds.nc, do.return = T, features.plot = "nGene", cols.use = c("grey",
    "green"))

plot_grid(p1, p2, p3[[1]], p4[[1]], ncol = 2, labels = "AUTO", label_size = 20, align = "h")
```

Integrating cells

We perform a canonical correlation analysis to integrate cells from different treatments. We combine cells collected using FACS and cells that had not been sorted and treat them as separate batches. Gene loadings for canonical components (CC) were used to identify "batch" genes and to curate the list of variable genes considered in URD trajectory reconstructions for interstitial cells.

```
# We follow https://satijalab.org/seurat/immune_alignment.html

# Getting cells from each batch.
p1 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "01-D1"]
p2 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "01-P2"]
p3 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "02-CO"]
p4 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "02-P1"]
p5 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "02-PB"]
p6 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "03-KI"]
p7 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "03-MA"]
p8 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "03-FM"]
p9 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "06-FM"]
p10 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "06-MA"]
p11 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "06-KI"]
p12 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "11-PO"]
p13 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "11-BU"]
p14 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "12-N1"]
p15 <- rownames(ds.nc@meta.data)[ds.nc@meta.data[, "orig.ident"] == "12-N2"]

# Nonfacs cells
nofa <- c(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13)
# Facs cells
fa <- c(p14, p15)

nofacs <- SubsetData(object = ds.nc, cells.use = nofa, subset.raw = TRUE)
facs <- SubsetData(object = ds.nc, cells.use = fa, subset.raw = TRUE)

nofacs@meta.data$batch <- "nofacs"
nofacs <- FilterCells(nofacs, subset.names = "nGene", low.thresholds = 300, high.thresholds = Inf)
nofacs <- NormalizeData(nofacs)
nofacs <- ScaleData(nofacs, display.progress = F)

facs@meta.data$batch <- "facs"
facs <- FilterCells(facs, subset.names = "nGene", low.thresholds = 300, high.thresholds = Inf)
facs <- NormalizeData(facs)
facs <- ScaleData(facs, display.progress = F)
```

```
# Gene selection for input to CCA
nofacs <- FindVariableGenes(nofacs, do.plot = F)
facs <- FindVariableGenes(facs, do.plot = F)
g.1 <- head(rownames(nofacs@hvg.info), 1000)
g.2 <- head(rownames(facs@hvg.info), 1000)
genes.use <- unique(c(g.1, g.2))
genes.use <- intersect(genes.use, rownames(nofacs@scale.data))
genes.use <- intersect(genes.use, rownames(facs@scale.data))
```

```
# Perform a canonical correlation analysis
hydra.combined <- RunCCA(nofacs, facs, genes.use = genes.use, num.cc = 30)
```

We visualize the CCA results and plot CC1 vs CC2 (Fig. 3)

```
# Visualize results of CCA plot CC1 versus CC2 and look at a violin plot.
p1 <- DimPlot(object = hydra.combined, reduction.use = "cca", group.by = "batch",
    pt.size = 0.5, do.return = TRUE)
p2 <- VlnPlot(object = hydra.combined, features.plot = "CC1", group.by = "batch",
    do.return = TRUE)
plot_grid(p1, p2)
```

```
# Explore gene loadings
PrintDim(object = hydra.combined, reduction.type = "cca", dims.print = 1:2, genes.print = 100)
```

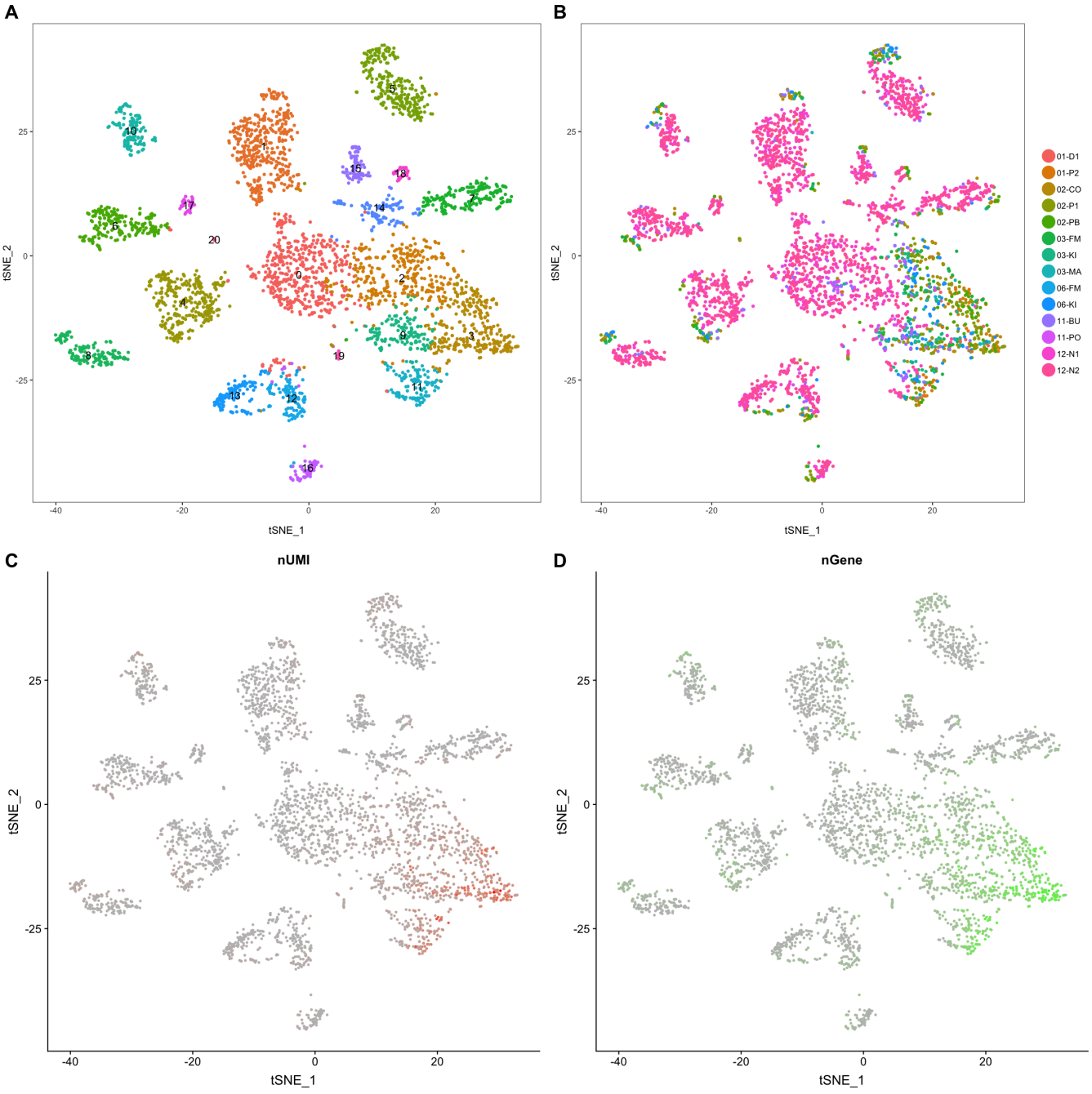

Figure 2: FACS introduces variability. A) t-SNE representation for all neuronal cells. B) Coloring by library reveals FACS batch effects. Cells from libraries 12- were sorted. C) Number of UMIs detected per cell. D) Number of genes detected per cell.

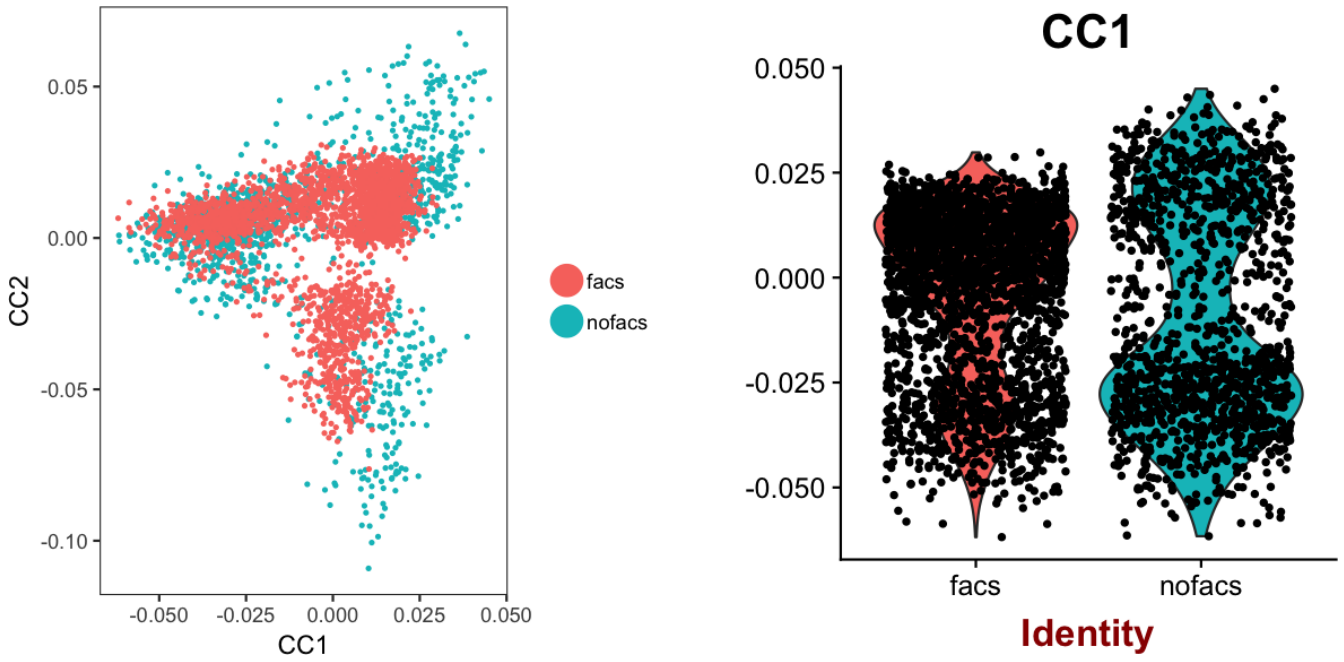

Figure 3: CCA results visualized. CC1 and CC2 separate cells the two batches. A) CC1 versus CC2 B) Violin plots comparing FACS and non-FACS cells.

We chose CCs 1-20 for the analysis (Fig. 4).

```
# Explores CCs
p3 <- MetageneBicorPlot(hydra.combined, grouping.var = "batch", dims.eval = 1:30,
    display.progress = FALSE)
```

```
# Explore CC heatmaps
DimHeatmap(object = hydra.combined, reduction.type = "cca", cells.use = 100, dim.use = 1,
    do.balanced = TRUE, margins = c(12, 16))
```

```
# Align subspaces
hydra.combined <- AlignSubspace(hydra.combined, reduction.type = "cca", grouping.var = "batch",
    dims.align = 1:20)

# p1 <- VlnPlot(object = hydra.combined, features.plot = 'ACC1', group.by =
# 'batch', do.return = TRUE) p2 <- VlnPlot(object = hydra.combined, features.plot
# = 'ACC2', group.by = 'batch', do.return = TRUE) plot_grid(p1, p2)
```

We run a single integrated analysis on all cells and visualize the results (Fig. 5).

```
# Perform integrated analysis
hydra.combined <- RunTSNE(hydra.combined, reduction.use = "cca.aligned", dims.use = 1:20,
    do.fast = T)

hydra.combined <- FindClusters(hydra.combined, reduction.type = "cca.aligned", resolution = 0.6,
    dims.use = 1:20)

# saveRDS('objects/ds.nc.integrated.rds')
```

```
# Load object from original analysis
hydra.combined <- readRDS("objects/ds.nc.integrated.rds")

p1 <- TSNEPlot(hydra.combined, do.label = T, do.return = T, pt.size = 0.5)
p2 <- TSNEPlot(hydra.combined, do.return = T, pt.size = 0.5, group.by = "batch")

plot_grid(p1, p2, ncol = 2, labels = "AUTO", label_size = 20, align = "h")
```

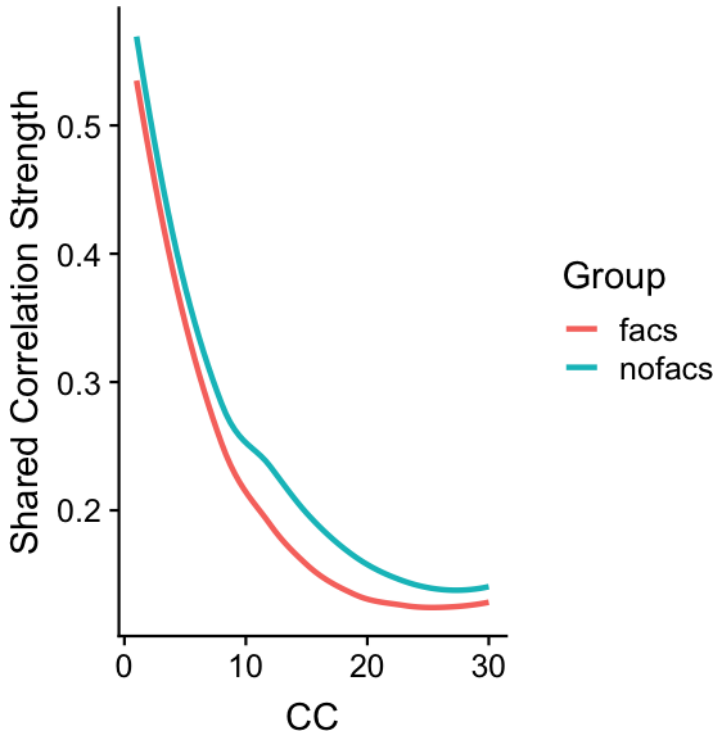

Figure 4: Examining CC correlation strength. Drop-off in signal is observed after CC20.

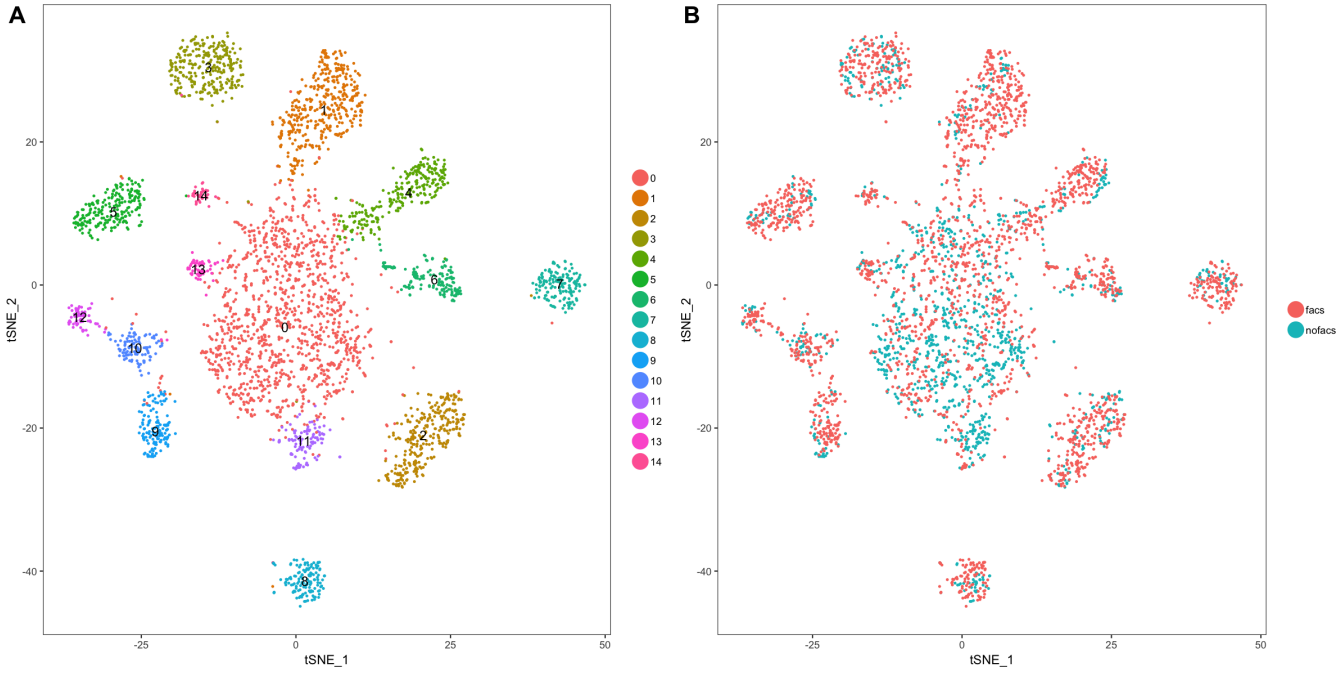

Figure 5: t-SNE representation after cell integration. A) t-SNE representation of neuronal cell clusters. B) Cells labeled by batch - FACS vs non-FACS.

Identify genes with differential expression in epithelia

The simple *Hydra* nervous system is composed of two nerve nets. One resides within the endoderm, the other within the ectoderm. We identify 12 neuronal clusters indicating distinct neuronal cell states. We are interested if these states can be assigned to a particular nerve net. We separate endoderm from ectoderm using body column tissue after removing head and foot. In this progress the neuronal cells stay associated with the respective epithelium. We assess differential gene expression using the package edgeR (*2*). We use the differentially expressed genes (that include differentially expressed neuronal genes) to place neuronal cells in one of the two epithelia. Three replicates each were collected for endodermal and ectodermal epithelial tissue. Outlier examination confirms the paired nature of the samples (Fig. 6).

```
# Tissue DGE

# Import endo ecto counts, three replicates for each treatment. Counts were
# generated using RSEM/Bowtie. s <-
# read.table('dge/DGE_ecto_endo_aepLRv2.matrix', header = TRUE)

s <- read.csv("objects/GSE121617_DGE_ecto_endo_aepLRv2.txt", header = TRUE, sep = "\t")

# Create ID column
s$ID <- rownames(s)

# Order columns
l <- s[c(7, 1:6)]

# p-value cutoff for significance used in all analyses
p.value = 0.05

###### endo ecto DGE Analysis#######

# Define treatment groups for analysis
TR <- factor(c("ec", "ec", "ec", "en", "en", "en"))
# Set wild type as referemce
TR <- relevel(TR, ref = "en")
# Create data frame defining treatment type for each sample
m <- data.frame(Sample = colnames(l[, c(2:7)]), TR)
# Define count columns
counts <- l[, c(2:7)]

# Fit model manually
y <- DGEList(counts = counts, group = m$TR, genes = s[, 1])

# Label each column with the appropriate sample name
colnames(y) <- m$Sample

# Calculate library size for each sample and store within DGElist
y$samples$lib.size <- colSums(y$counts)
# Exclude transcripts that do not have at least two samples with more than one
# count per million
keep <- rowSums(cpm(y) > 1) >= 2
y <- y[keep, , keep.lib.sizes = FALSE]

# Calculate normalization factors for each sample
y <- calcNormFactors(y)

# review library sizes and normalization factors y$samples

# Create a matrix defining the control and treatment groups for the analysis
# based on what is described by the TR object
design <- model.matrix(~TR)

# Estimate dispersions
y <- estimateDisp(y, design)

# Explore outliers
plotMDS(y, method = "bcv", cex = 1)

# Get normalized counts, apply_normalizations() from package agalmar
norm <- apply_normalizations(y)
norm <- as.data.frame(round(norm, digits = 0))
# Rename colums
```

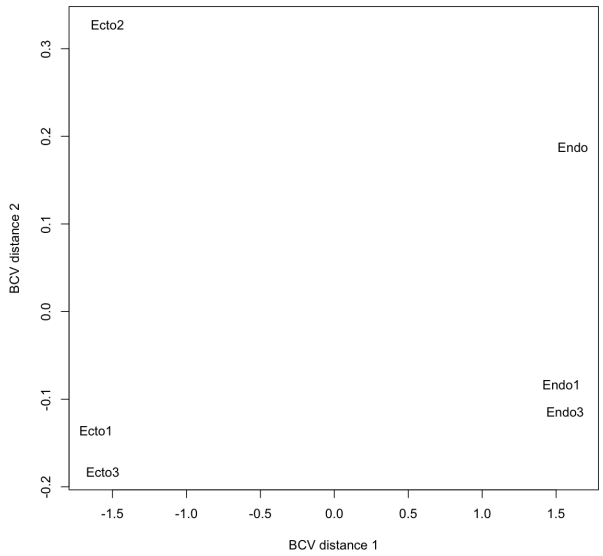

Figure 6: MDS plot for epithelial libraries. Distances between samples correspond to the leading biological coefficient of variation (BCV).

```
colnames(norm) <- paste("n", colnames(norm), sep = "")
# Restore ID column
norm$ID <- rownames(norm)

# get normalized counts using DGElist object norm <- get_norm(y)

# Merge normalized counts to main dataframe
f <- merge(l, norm, by = "ID")

# Fit data to a negative binomial generalized linear model
fit <- glmFit(y, design)

# Conduct a statistical test for differential gene expression based on the fit
lrt <- glmLRT(fit)

# Create a list of values that describes the status of each gene in the DGE test
de <- decideTestsDGE(lrt, adjust.method = "BH", p.value)

# Overview of number of differentially expressed genes
sum <- summary(de)

detags <- rownames(y)[as.logical(de)]
plotSmear(lrt, de.tags = detags, cex = 0.2, cex.lab = 1, cex.axis = 1)
abline(h = c(-1, 1), col = "blue")
```

We obtained expression data for 13995 genes. 3055 genes are found to be significantly differentially expressed in the endodermal epithelium; 2859 genes are found to be significantly differentially expressed in the ectodermal epithelium (adjusted p-value <0.05) (Fig. 7).

Placement of neuronal cell states in epithelia within the body column

We use the Seurat function AddModuleScore to score each cell in the neuronal dataset for sets of genes specific to the endodermal epithelium or to the ectodermal epithelium respectively (Fig. 8).

```
# Build results table
D <- lrt$table

# Restore ID column
D$ID <- rownames(D)

# Add column of adjusted p values
D <- cbind(D, p.adjust(D$PValue, method = "BH"))
```

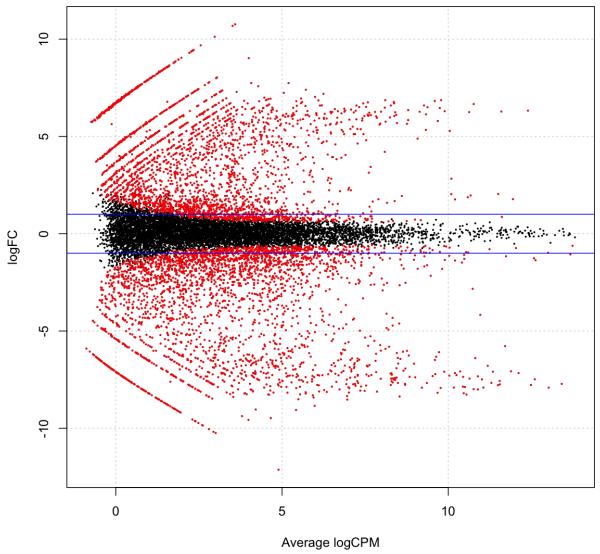

Figure 7: Differential gene expression test results. Genes that are significantly differentially expressed are colored in red. Positive fold-change indicates enrichment in the endodermal epithelium. x-axis shows log-scaled abundance (counts per million).

```
names(D)[names(D) == "p.adjust(D$PValue, method = \"BH\")"] = "Padj"

res <- merge(f, D, by = "ID")

write.csv(res, "objects/endo_ecto_dge.csv")

ecto_sig <- subset(res, res$Padj <= 0.05 & res$logFC > 2)
write.csv(ecto_sig, "objects/ecto_transcripts.csv")

endo_sig <- subset(res, res$Padj <= 0.05 & res$logFC < -2)
write.csv(endo_sig, "objects/endo_transcripts.csv")
```

For the scoring we consider genes that show a fold-change > 2 (1797 ectodermal genes, 1715 endodermal genes).

```
# Placing cells
ecto_sig <- read.csv("objects/ecto_transcripts.csv")
endo_sig <- read.csv("objects/endo_transcripts.csv")

# Pull ids for genes diffentially expressed in ectoderm
Ecto <- ecto_sig$ID
# Pull ids for genes diffentially expressed in endoderm
Endo <- endo_sig$ID

# Calculate endo and ecto module scores for ds data
hydra.combined <- AddModuleScore(hydra.combined, genes.list = list(Endo = Endo, Ecto = Ecto))

# nGene nUMI orig.ident percent.mito res.1.5 cluster_numbering stim res.0.6
# Cluster1 Cluster2
colnames(hydra.combined@meta.data) <- c("nGene", "nUMI", "orig.ident", "percent.mito",
    "res.1.5", "cluster_numbering", "batch", "res.0.6", "Endo", "Ecto")

# saveRDS(hydra.combined,'objects/exploration/ds.nc.integrated_tissue.rds')
```

```
# Load object from original analysis
hydra.combined <- readRDS("objects/ds.nc.integrated_tissue.rds")

# Plot tSNE and clusters labeled epithelial origin
p1 <- TSNEPlot(hydra.combined, do.label = T, do.return = T, pt.size = 0.5, no.legend = TRUE)
p2 <- FeaturePlot(hydra.combined, c("Ecto", "Endo"), overlay = T, cols.use = c("grey",
    "blue", "red"), max.cutoff = 0.1, do.return = TRUE)
```

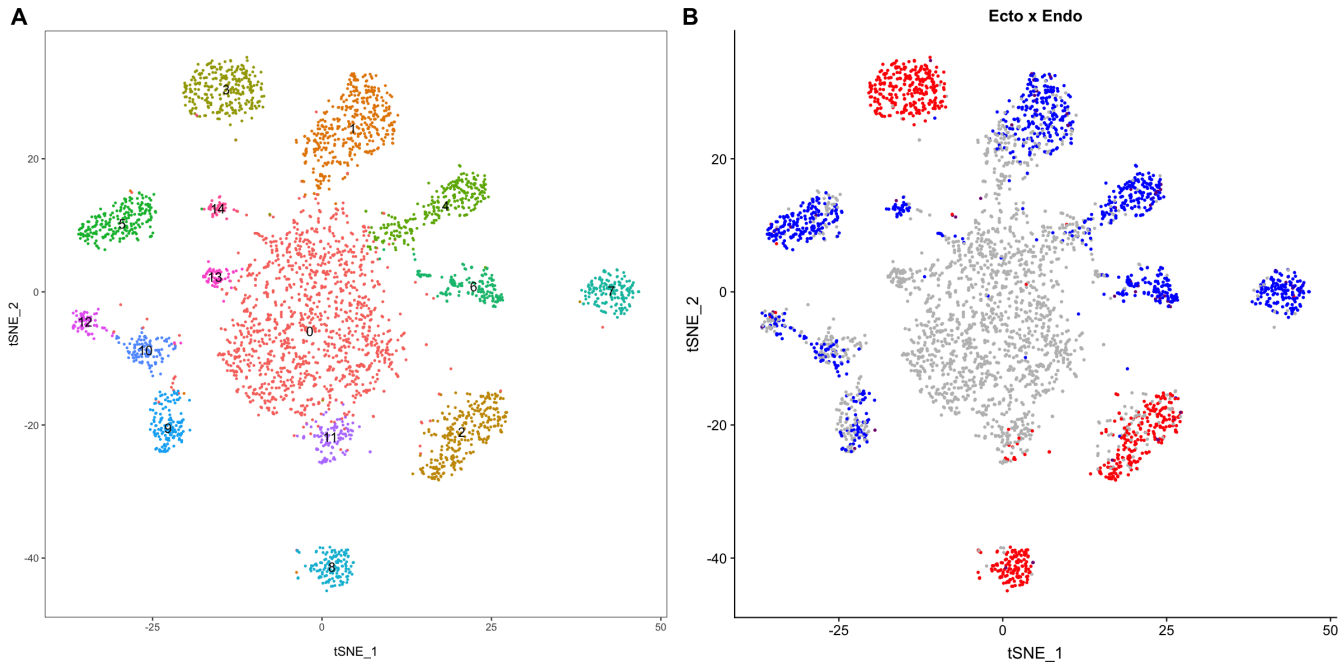

Figure 8: Placing neuronal cell states in epithelia. Scoring cells for genes specific to epithelia suggest that cells in clusters 2,3, and 8 originated from the endodermal epithelium. Cells in cluster 11 are putative endodermal gland and nerve cell progenitors. blue: high scores for ectodermal gene set, red: high scores for endodermal gene set.

```
plot_grid(p1, p2[[1]], ncol = 2, labels = "AUTO", label_size = 20, align = "h")
```

```
# find markers for all clusters

nc.markers <- FindAllMarkers(object = hydra.combined, only.pos = TRUE, min.pct = 0.25,
    thresh.use = 0.25)

top12 <- nc.markers %>% group_by(cluster) %>% top_n(12, avg_logFC)
DoHeatmap(object = hydra.combined, genes.use = top12$gene, slim.col.label = TRUE,
    remove.key = TRUE)
```

```
# saveRDS(hydra.combined,'objects/exploration/ds.nc.integrated_tissue_markers.rds')

# pdf('mFig/neuro_marker.pdf', width=21,height=19, bg ='white',
# useDingbats=FALSE) DoHeatmap(object = hydra.combined, genes.use = top12$gene,
# slim.col.label = TRUE, remove.key = TRUE) dev.off()
```

Annotate clusters

We annotate neuronal clusters using known and newly identified neuronal markers (Figs. 10, 11).

```
# Plot for known and new neuronal markers

hydra.combined <- readRDS("objects/ds.nc.integrated_tissue.rds")

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

# Use this function to find the full ID for your gene of interest
hFind <- function(x) {
    return(hydra.combined@data@Dimnames[[1]][grep(x, hydra.combined@data@Dimnames[[1]],
```

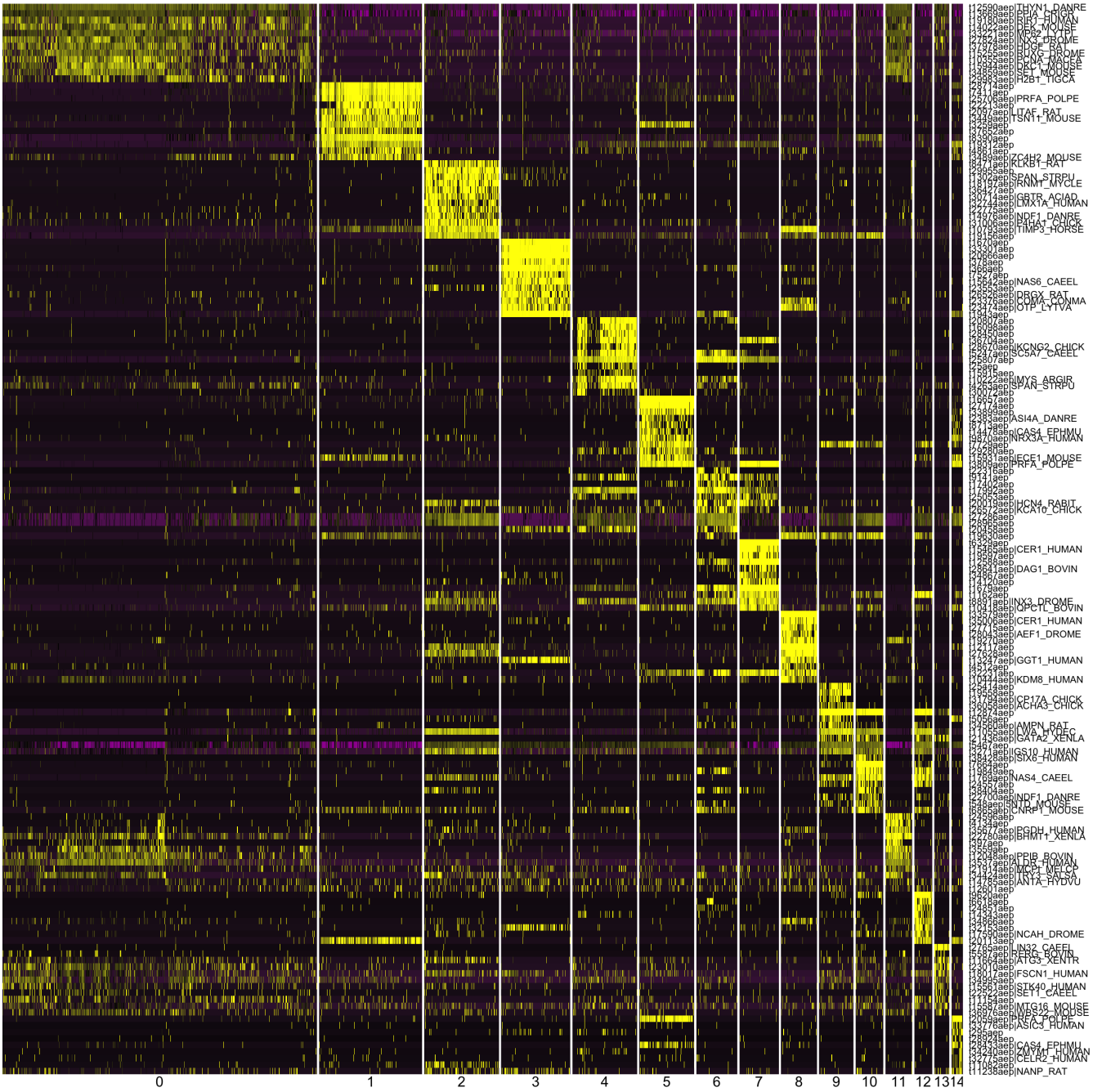

Figure 9: Markers (top 12) for neuronal subtypes.

```
        ignore.case = T)])
}

p <- TSNEPlot(hydra.combined, do.label = T, do.return = T, pt.size = 1, label.size = 8,
    no.legend = TRUE)
p1 <- FeaturePlot(hydra.combined, c("Ecto", "Endo"), overlay = T, cols.use = c("grey",
    "blue", "red"), max.cutoff = 0.1, do.return = TRUE)

gene.names <- c(hFind("t1679aep"), hFind("t8891aep"), hFind("t3809aep"), hFind("t2059aep"),
    hFind("t25706aep"), hFind("t16657aep"), hFind("t33899aep"), hFind("t11055aep"),
    hFind("t12874aep"), hFind("t10853aep"), hFind("t1163aep"), hFind("t20256aep"),
    hFind("t5467aep"), hFind("t3974aep"), hFind("t6329aep"), hFind("t28450aep"),
    hFind("t14976aep"), hFind("t33301aep"))

new.names <- c("Hym-176 A", "Innexin 2", "prec A", "prec B", "prec C", "prec D",
    "prec E", "LW-amide", "Hym-355", "CnASH", "Cnot", "prdl-a", "NDA-1", "ELAV2 (t3974)",
    "clust7 marker (t6329)", "clust4 marker (t28450)", "NDF1 (t14976)", "Alpha-LTX-Lhe1a-like (t33301)")

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

p2 <- FeaturePlot(hydra.combined, c("Hym-176 A", "Innexin 2", "prec A", "prec B",
    "prec C", "prec D", "prec E", "LW-amide", "Hym-355", "CnASH"), cols.use = c("grey",
    "blue"), do.return = TRUE)

plot_grid(p, p1[[1]], p2[[1]], p2[[2]], p2[[3]], p2[[4]], p2[[5]], p2[[6]], p2[[7]],
    p2[[8]], p2[[9]], p2[[10]], labels = "AUTO", label_size = 30, align = "h", ncol = 4)
```

```
p2 <- FeaturePlot(hydra.combined, c("Cnot", "prdl-a", "NDA-1", "ELAV2 (t3974)", "clust7 marker (t6329)",
    "clust4 marker (t28450)", "NDF1 (t14976)", "Alpha-LTX-Lhe1a-like (t33301)"),
    cols.use = c("grey", "blue"), do.return = TRUE)

plot_grid(p2[[1]], p2[[2]], p2[[3]], p2[[4]], p2[[5]], p2[[6]], p2[[7]], p2[[8]],
    labels = "AUTO", label_size = 30, align = "h", ncol = 4)
```

```
hydra.combined <- readRDS("objects/ds.nc.integrated_tissue.rds")

# LABELING neuro cluster Store cluster numbering
hydra.combined <- StashIdent(object = hydra.combined, save.name = "cluster_numbering")

current.cluster.ids <- as.character(0:14)

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

cluster.names <- c("0_prog", "1_S_ect_tent", "2_G_end_1", "3_S_end_2", "4_G_ect_tent/head",
    "5_G_ect_tent", "6_G_ect_body", "7_G_ect_ped", "8_unkown_end_3", "9_G_ect_head/tent",
    "10_G_ect_body", "11_endo_gc_nc_prog", "12_G_ect_bd", "13_nc_prog", "14_S_ect_head/hyp")

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

TSNEPlot(object = hydra.combined, do.return = T, do.label = T, label.size = 6, no.legend = TRUE)
```

Characterization of the innexin gene family

Innexins are transmembrane proteins that form gap junctions in invertebrates. The *Hydra* genome encodes a family of 17 innexin genes (*12*). The expression of two of these genes has been characterized. Innexin 1 has been shown to form gap junctions in ectodermal cells (*13*). Neuronal expression and a role in *Hydra* contractile behavior has been reported for innexin 2 (*4*). We analyze innexin expression in all cells of the data set and the neuronal subset (Fig. 13, 14). Drop-seq data reveal a broad diversity of expression patterns and roles of innexins in different cell lineages and subsets of cells.

```
# We explore expression of innexins in all cells of the data set

# Load object that includes all cells
ds.s1 <- readRDS("objects/Hydra_Seurat_Whole_Transcriptome.rds")

## Suspected doublet cluster db (68 cells) was excluded from from downstream
```

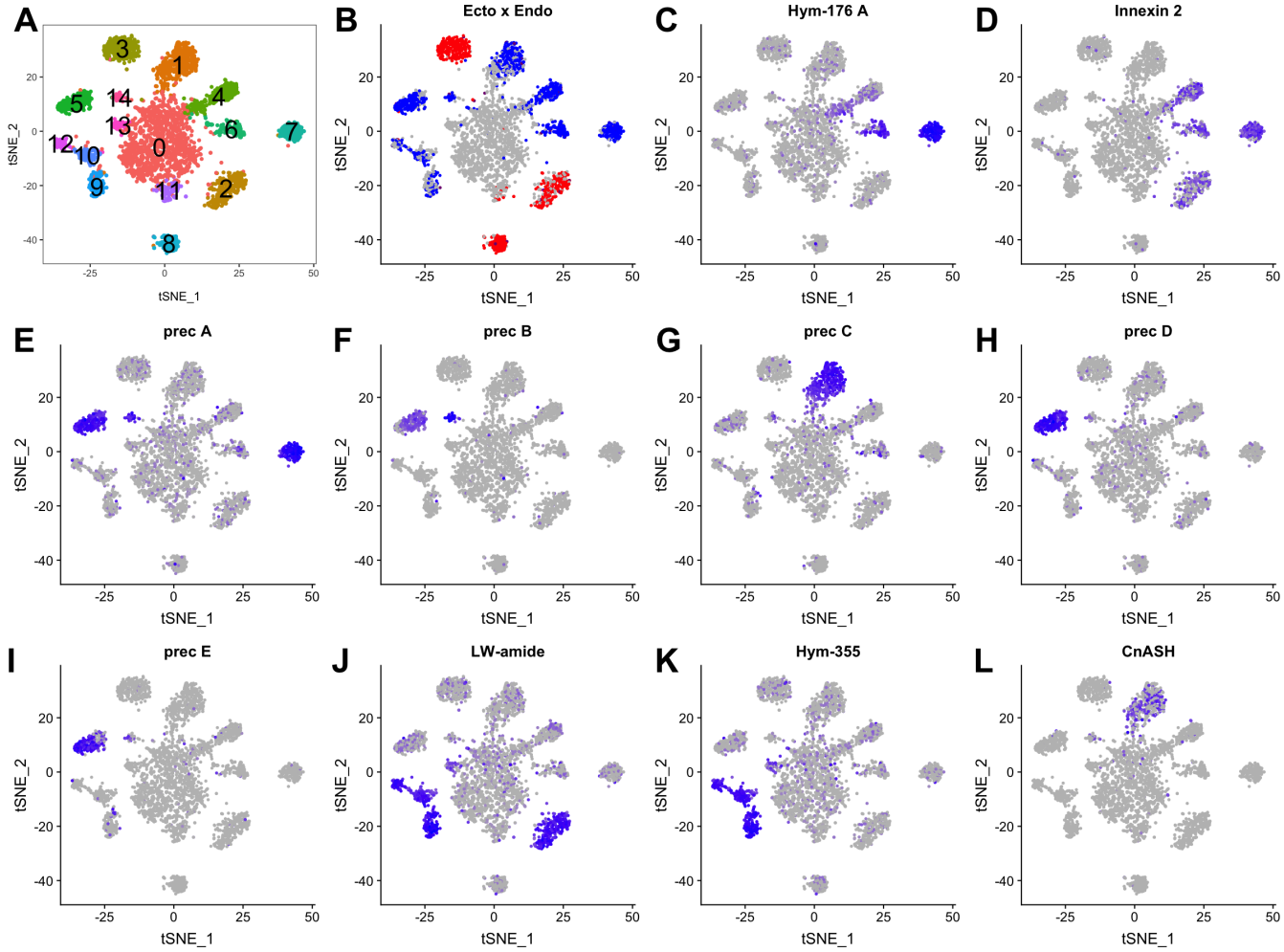

Figure 10: Expression of known and newly (without reference) identified neuronal markers used in neuronal cluster annotation. A) Neuronal subclustering. B) Neuronal cells colored by epithelial origin. Red - endoderm, blue ectoderm. C) *Hym-176 A*, ectodermal ganglion neurons (head/tentacles, body column, peduncle) (*3*). D) *Innexin 2*, ectodermal ganglion cells (peduncle) (*4*). Data suggests wider expression in ectodermal neurons of body column, head and a subset of endodermal neurons. E) *RFamide preprohormone A*, ectodermal sensory and ganglion neurons (tentacle, head/hypostome, peduncle) (*5*). F) *RFamide preprohormone B*, ectodermal sensory neurons (tentacle, head/hypostome) (*5*). G) *RFamide preprohormone C*, ectodermal sensory neurons (tentacles) (*5*). H) *RFamide preprohormone D*, ectodermal sensory neurons (tentacles) (*6*). I) *RFamide preprohormone E*, ectodermal sensory neurons (tentacles) (this study). J) *LW-amide*, ectodermal ganglion neurons (tentacles/head, body column, and basal disk, subset of endodermal neurons) (this study) (*7*). K) *Hym-355*, ectodermal ganglion neurons (tentacles/head, body column, and basal disk) (*8*). L) *CnASH*, ectodermal sensory neurons (tentacles) (*6*).

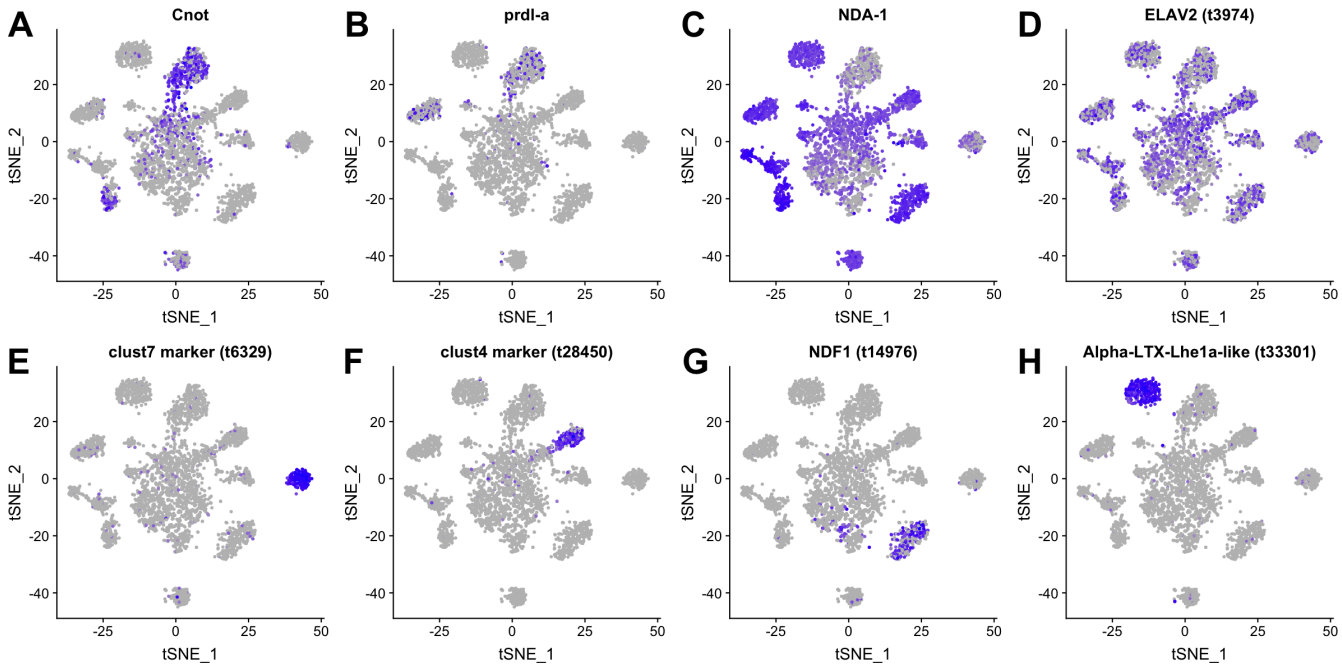

Figure 11: Expression of known and newly (without reference) identified neuronal markers used in neuronal cluster annotation. A) *Cnot*, ectodermal sensory neurons (tentacles) (*9*). Data suggest a broader expression in a subset of ectodermal ganglion neurons and a subset of endodermal neurons. B) *prdl-a*,ectodermal sensory neurons (tentacles) (*10*). C) *NDA-1*, sensory and ganglion neurons throughout body (*11*). Endodermal neurons (this study). D) *ELAV2* (t3974), all neuronal clusters and in neuronal progenitors (this study). E) *t6329*, peduncle neurons (this study). F) *t28450*, tentacle neurons (this study). G) *NDF1* (t14976), endodermal ganglion neurons (this study). H) *Alpha-LTX-Lhe1a-like* (t33301), endodermal sensory neurons (this study).

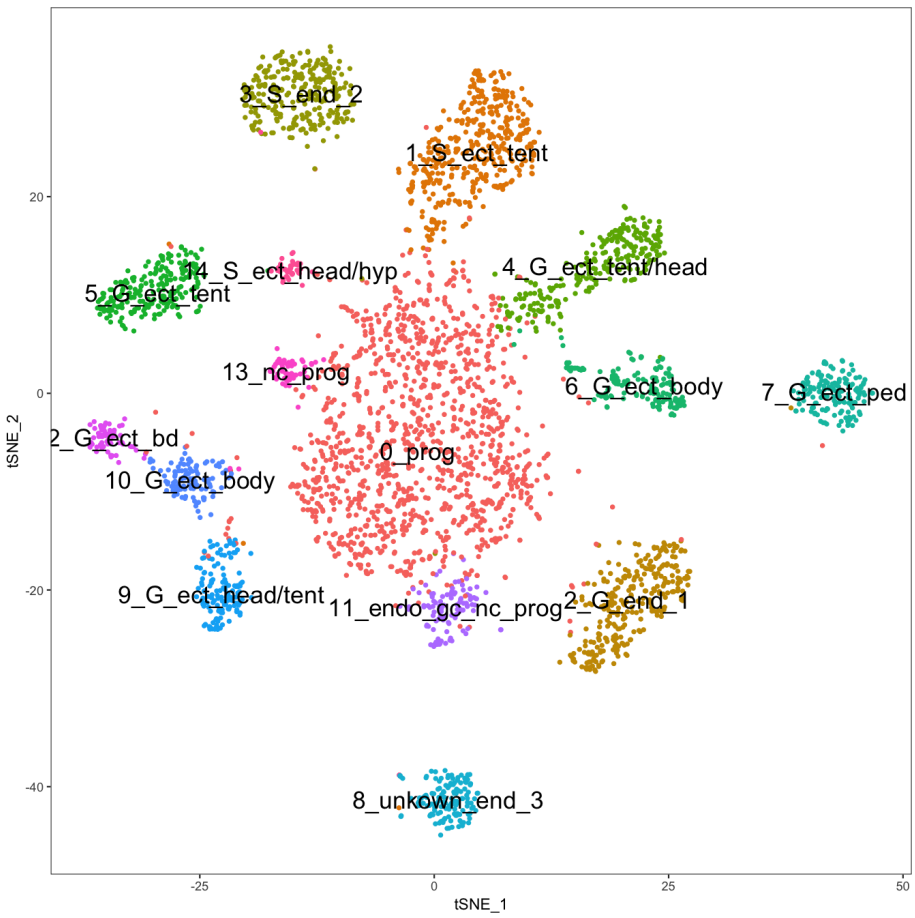

Figure 12: Neuronal cell states labeled with cluster id, tentative neuron class based on published or new findings, in vivo localisation in the animal based on published or new findings. bd: basal disk, ect: ectoderm, end: endoderm, G: ganglion neuron, gc: gland cell, nc: neuronal cell, ped: peduncle, prog: progenitor, S: sensory neuron, tent: tentacle.

```
## analyses
ds.s1 <- SubsetData(object = ds.s1, ident.remove = c("db"), subset.raw = TRUE)

# Function to annotate/rename genes
update.names <- function(gene.names, new.names) {
    for (i in 1:length(gene.names)) {
        rownames(ds.s1@data)[which(rownames(ds.s1@data) == gene.names[i])] <<- new.names[i]
    }
}

# Use this function to find the full ID for your gene of interest
hFind <- function(x) {
    return(ds.s1@data@Dimnames[[1]][grep(x, ds.s1@data@Dimnames[[1]], ignore.case = T)])
}

gene.names <- c(hFind("t4922aep"), hFind("t27824aep"), hFind("t8891aep"), hFind("t28211aep"),
    hFind("t7498aep"), hFind("t25468aep"), hFind("t24557aep"), hFind("t34437aep"),
    hFind("t23010aep"), hFind("t8479aep"), hFind("t21368aep"), hFind("t13971aep"),
    hFind("t10604aep"), hFind("t4858aep"), hFind("t15539aep"), hFind("t27709aep"),
    hFind("t21244aep"))

new.names <- c("Innexin 1", "Innexin 1A", "Innexin 2", "Innexin 3", "Innexin 4",
    "Innexin 5", "Innexin 6", "Innexin 7", "Innexin 8", "Innexin 9", "Innexin 9a",
    "Innexin 10", "Innexin 11", "Innexin 12", "Innexin 13", "Innexin 14", "Innexin 15")

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

p2 <- FeaturePlot(ds.s1, c("Innexin 1", "Innexin 1A", "Innexin 2", "Innexin 3", "Innexin 4",
    "Innexin 5", "Innexin 6", "Innexin 7", "Innexin 8", "Innexin 9", "Innexin 9a",
    "Innexin 10", "Innexin 11", "Innexin 12", "Innexin 13", "Innexin 14", "Innexin 15"),
    cols.use = c("grey", "blue"), do.return = TRUE)

# plot_grid(p2[[1]],p2[[2]],p2[[3]],p2[[4]],p2[[5]],p2[[6]],p2[[7]],p2[[8]],p2[[9]],p2[[10]],p2[[11]],p2[[
# labels='AUTO', label_size = 20, align = 'h', ncol=4)

plot_grid(plotlist = p2, labels = "AUTO", label_size = 20, align = "h", ncol = 4)
```

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

# use this function to find the full ID for your gene of interest
hFind <- function(x) {
    return(hydra.combined@data@Dimnames[[1]][grep(x, hydra.combined@data@Dimnames[[1]],
        ignore.case = T)])
}

gene.names <- c(hFind("t4922aep"), hFind("t27824aep"), hFind("t8891aep"), hFind("t28211aep"),
    hFind("t7498aep"), hFind("t25468aep"), hFind("t24557aep"), hFind("t34437aep"),
    hFind("t23010aep"), hFind("t8479aep"), hFind("t21368aep"), hFind("t13971aep"),
    hFind("t10604aep"), hFind("t4858aep"), hFind("t15539aep"), hFind("t27709aep"),
    hFind("t21244aep"))

new.names <- c("Innexin 1", "Innexin 1A", "Innexin 2", "Innexin 3", "Innexin 4",
    "Innexin 5", "Innexin 6", "Innexin 7", "Innexin 8", "Innexin 9", "Innexin 9a",
    "Innexin 10", "Innexin 11", "Innexin 12", "Innexin 13", "Innexin 14", "Innexin 15")

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

p2 <- FeaturePlot(ds.s1, c("Innexin 1", "Innexin 1A", "Innexin 2", "Innexin 3", "Innexin 4",
    "Innexin 5", "Innexin 6", "Innexin 7", "Innexin 8", "Innexin 9", "Innexin 9a",
    "Innexin 10", "Innexin 11", "Innexin 12", "Innexin 13", "Innexin 14", "Innexin 15"),
    cols.use = c("grey", "blue"), do.return = TRUE)

# plot_grid(p2[[1]],p2[[2]],p2[[3]],p2[[4]],p2[[5]],p2[[6]],p2[[7]],p2[[8]],p2[[9]],p2[[10]],p2[[11]],p2[[
# labels='AUTO', label_size = 20, align = 'h', ncol=4)

plot_grid(plotlist = p2, labels = "AUTO", label_size = 20, align = "h", ncol = 4)
```

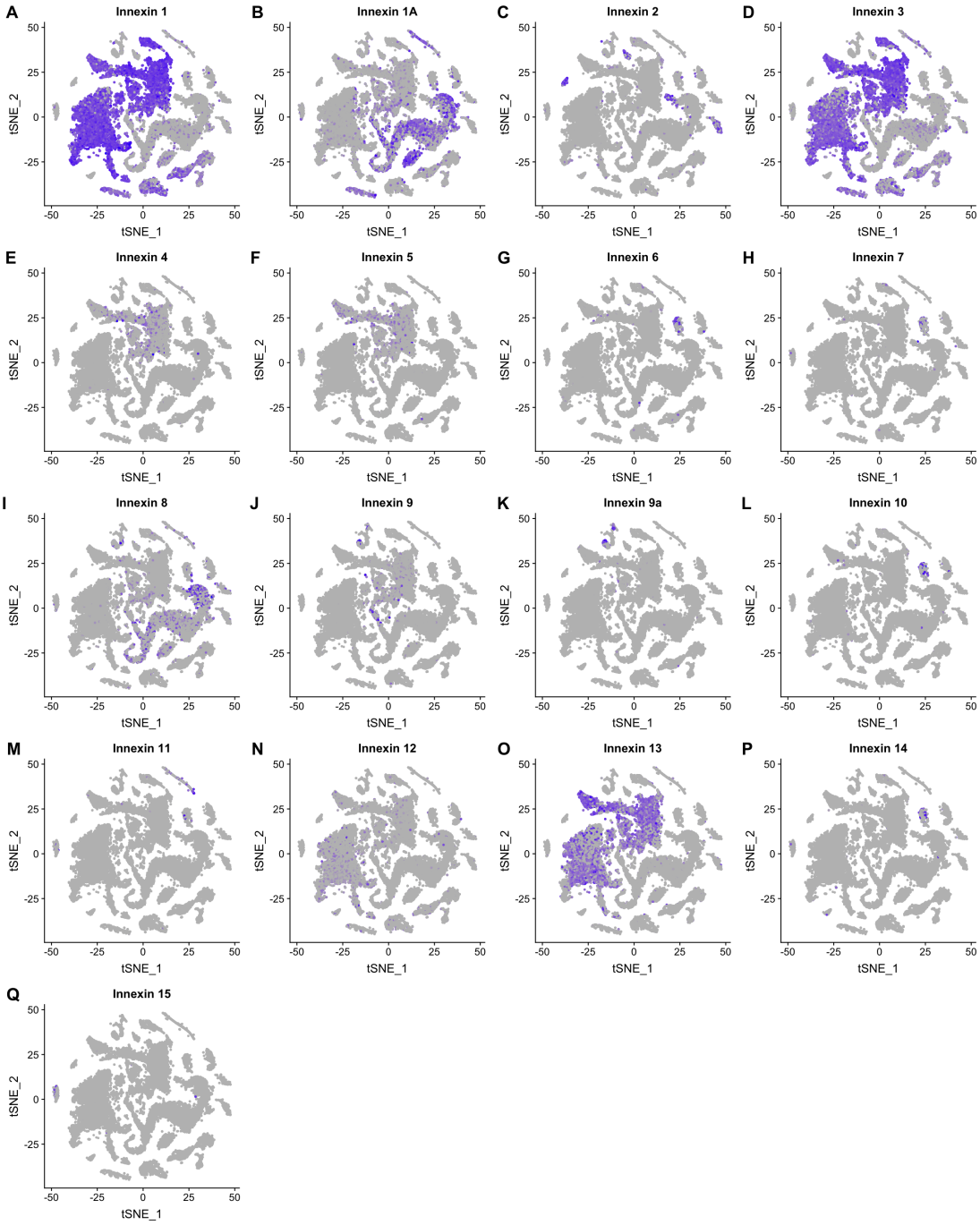

Figure 13: Innexin family members with cell lineage and cell state specific expression. Innexin expression visualized on the full t-SNE. A) Innexin1 - endodermal and ectodermal epithelial cells, gland cells. B) Innexin 1a - interstitial stem cells, progenitor cells, female and male germline. C) Innexin2 - subset of differentiated neurons. D) Innexin3 - endodermal and ectodermal epithelial cells, gland cells. E) Innexin 4 - ectodermal epithelial cells. F) Innexin 5 - ectodermal epithelial cells. G) Innexin 6 - specific neuron. H) Innexin 7 - specific neuron. I) Innexin 8 - differentiating interstitial cells, neurons. J) Innexin 9 - nematoblast subpopulation. K) Innexin 9a - differentiated nematocytes. L) Innexin 10 - specific neuron. M) Innexin 11 - male germline. N) Innexin 12 - endodermal epithelial cells, neuron. O) Innexin 13 - endodermal and ectodermal epithelial cells. P) Innexin 14 - specific neuron. Q) Innexin 15 - specific neuron.

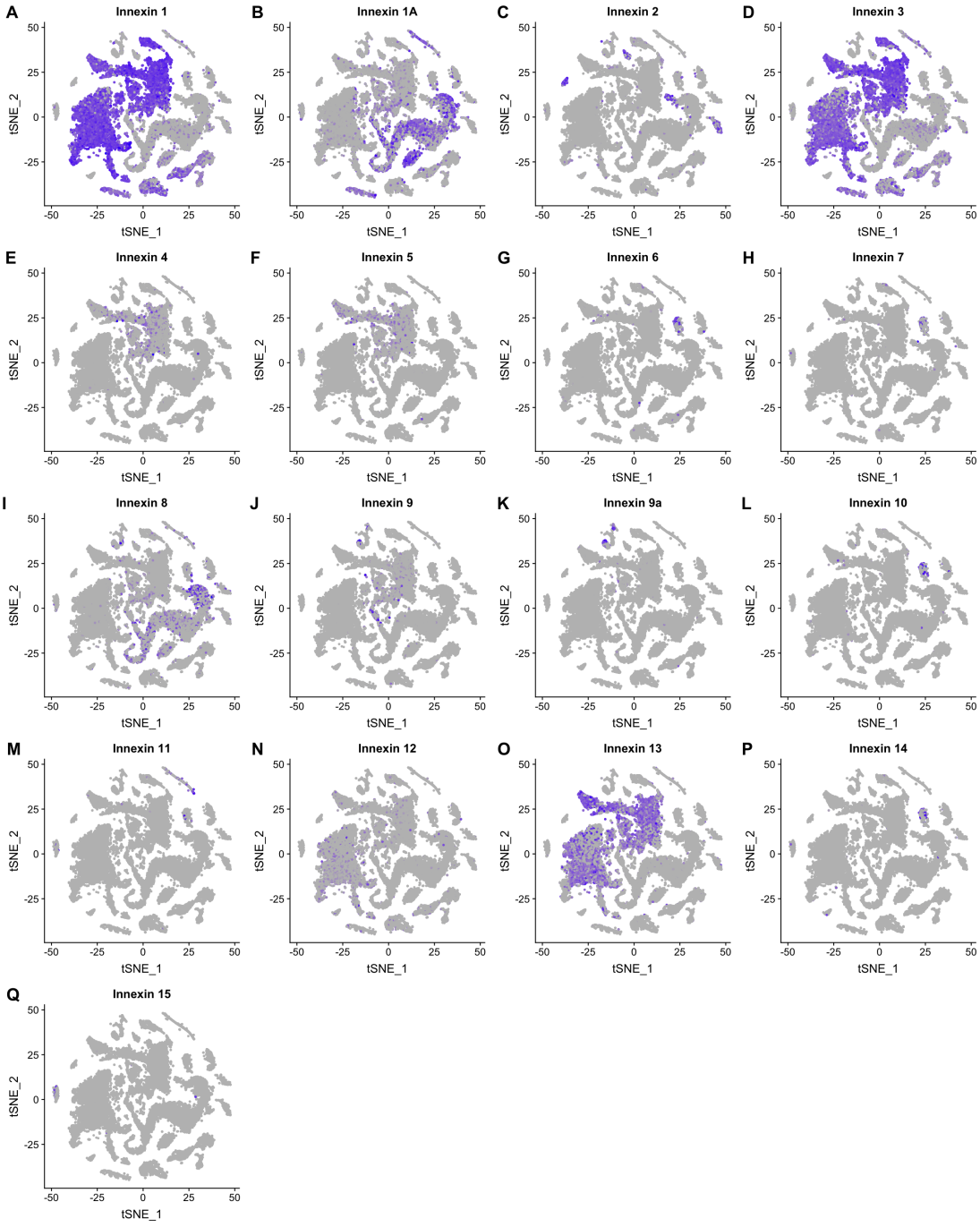

Figure 14: Innexin family members with cell lineage and cell state specific expression. Innexin expression visualized on the full t-SNE. A) Innexin1 - endodermal and ectodermal epithelial cells, gland cells. B) Innexin 1a - interstitial stem cells, progenitor cells, female and male germline. C) Innexin2 - subset of differentiated neurons. D) Innexin3 - endodermal and ectodermal epithelial cells, gland cells. E) Innexin 4 - ectodermal epithelial cells. F) Innexin 5 - ectodermal epithelial cells. G) Innexin 6 - specific neuron. H) Innexin 7 - specific neuron. I) Innexin 8 - differentiating interstitial cells, neurons. J) Innexin 9 - nematoblast subpopulation. K) Innexin 9a - differentiated nematocytes. L) Innexin 10 - specific neuron. M) Innexin 11 - male germline. N) Innexin 12 - endodermal epithelial cells, neuron. O) Innexin 13 - endodermal and ectodermal epithelial cells. P) Innexin 14 - specific neuron. Q) Innexin 15 - specific neuron.

Software versions

This document was computed on Thu Nov 01 16:16:00 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
 [4] gtable_0.2.0       dplyr_0.7.7        Seurat_2.2.1
 [7] Matrix_1.2-12      cowplot_0.9.2      ggplot2_3.1.0
[10] agalmar_0.0.0.9000 edgeR_3.20.9       limma_3.34.9
[13] knitr_1.20

loaded via a namespace (and not attached):
  [1] backports_1.1.2            Hmisc_4.1-1
  [3] VGAM_1.0-5                 sn_1.5-1
  [5] plyr_1.8.4                 igraph_1.2.2
  [7] lazyeval_0.2.1             splines_3.4.3
  [9] BiocParallel_1.12.0        GenomeInfoDb_1.14.0
 [11] digest_0.6.18              foreach_1.4.4
 [13] htmltools_0.3.6            lars_1.2
 [15] gdata_2.18.0               magrittr_1.5
 [17] checkmate_1.8.5            memoise_1.1.0
 [19] cluster_2.0.6              mixtools_1.1.0
 [21] ROCR_1.0-7                 sfsmisc_1.1-2
 [23] recipes_0.1.2              annotate_1.56.1
 [25] gower_0.1.2                dimRed_0.1.0
 [27] matrixStats_0.53.1         R.utils_2.6.0
 [29] colorspace_1.3-2           blob_1.1.0
 [31] xfun_0.1                   crayon_1.3.4
 [33] RCurl_1.95-4.10            genefilter_1.60.0
 [35] bindr_0.1.1                survival_2.41-3
 [37] iterators_1.0.9            ape_5.1
 [39] glue_1.3.0                 DRR_0.0.3
 [41] ipred_0.9-6                zlibbioc_1.24.0
 [43] XVector_0.18.0             DelayedArray_0.4.1
 [45] kernlab_0.9-25             ddalpha_1.3.1.1
 [47] prabclus_2.2-6             BiocGenerics_0.24.0
 [49] DEoptimR_1.0-8             scales_1.0.0
 [51] mvtnorm_1.0-7              DBI_0.8
 [53] Rcpp_0.12.19               dtw_1.18-1
 [55] metap_0.8                  xtable_1.8-2
 [57] htmlTable_1.11.2           tclust_1.3-1
 [59] proxy_0.4-21               foreign_0.8-69
 [61] bit_1.1-12                 mclust_5.4
 [63] SDMTools_1.1-221           Formula_1.2-2
 [65] tsne_0.1-3                 lava_1.6
 [67] prodlim_1.6.1              stats4_3.4.3
 [69] htmlwidgets_1.0            FNN_1.1
 [71] gplots_3.0.1               RColorBrewer_1.1-2
 [73] fpc_2.1-11                 acepack_1.4.1
 [75] modeltools_0.2-21          ica_1.0-1
 [77] pkgconfig_2.0.2            XML_3.98-1.10
 [79] R.methodsS3_1.7.1          flexmix_2.3-14
 [81] nnet_7.3-12                locfit_1.5-9.1
 [83] caret_6.0-78               labeling_0.3
 [85] tidyselect_0.2.5           reshape2_1.4.3
```

```
 [87] AnnotationDbi_1.40.0      munsell_0.5.0
 [89] tools_3.4.3               RSQLite_2.0
 [91] ranger_0.9.0              ggridges_0.4.1
 [93] broom_0.4.3               evaluate_0.10.1
 [95] stringr_1.3.0             yaml_2.1.18
 [97] ModelMetrics_1.1.0        bit64_0.9-7
 [99] robustbase_0.92-8         caTools_1.17.1.1
[101] purrr_0.2.5               pbapply_1.3-4
[103] nlme_3.1-131.1            formatR_1.5
[105] R.oo_1.21.0               RcppRoll_0.2.2
[107] compiler_3.4.3            rstudioapi_0.7
[109] tibble_1.4.2              geneplotter_1.56.0
[111] stringi_1.1.6             highr_0.6
[113] lattice_0.20-35           trimcluster_0.1-2
[115] psych_1.7.8               diffusionMap_1.1-0
[117] pillar_1.2.1              irlba_2.3.2
[119] data.table_1.11.8         bitops_1.0-6
[121] GenomicRanges_1.30.3      R6_2.3.0
[123] latticeExtra_0.6-28       bookdown_0.7
[125] KernSmooth_2.23-15        IRanges_2.12.0
[127] codetools_0.2-15          MASS_7.3-49
[129] gtools_3.5.0              assertthat_0.2.0
[131] CVST_0.2-1                SummarizedExperiment_1.8.1
[133] DESeq2_1.18.1             rprojroot_1.3-2
[135] withr_2.1.2               mnormt_1.5-5
[137] S4Vectors_0.16.0          GenomeInfoDbData_1.0.0
[139] diptest_0.75-7            parallel_3.4.3
[141] rpart_4.1-13              timeDate_3043.102
[143] tidyr_0.8.0               class_7.3-14
[145] rmarkdown_1.9             segmented_0.5-3.0
[147] Rtsne_0.13                numDeriv_2016.8-1
[149] lubridate_1.7.3           scatterplot3d_0.3-40
[151] Biobase_2.38.0            base64enc_0.1-3
```

References

1. A. Butler, P. Hoffman, P. Smibert, E. Papalexi, R. Satija, Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology.* **36**, 411–420 (2018).

2. Robinson, Mark D, McCarthy, Davis J, Smyth, Gordon K, edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics.* **26**, 139–140 (2010).

3. Yum, S *et al.*, A novel neuropeptide, Hym-176, induces contraction of the ectodermal muscle in Hydra. *Biochemical and biophysical research communications.* **248**, 584–590 (1998).

4. Y. Takaku *et al.*, Innexin gap junctions in nerve cells coordinate spontaneous contractile behavior in Hydra polyps. *Scientific reports.* **4**, 3573 (2014).

5. Darmer, D *et al.*, Three different prohormones yield a variety of Hydra-RFamide (Arg-Phe-NH2) neuropeptides in Hydra magnipapillata. *The Biochemical journal.* **332 (Pt 2)**, 403–412 (1998).

6. Hayakawa, Eisuke, Fujisawa, Chiemi, Fujisawa, Toshitaka, Involvement of Hydra achaete-scute gene CnASH in the differentiation pathway of sensory neurons in the tentacles. *Wilhelm Roux' Archiv für Entwicklungsmechanik der Organismen* (2004).

7. Mitgutsch, Christian, Hauser, Frank, Grimmelikhuijzen, Cornelis J P, Expression and developmental regulation of the Hydra-RFamide and Hydra-LWamide preprohormone genes in Hydra: Evidence for transient phases of head formation. **207**, 189–203 (1999).

8. Takahashi, T *et al.*, A novel neuropeptide, Hym-355, positively regulates neuron differentiation in Hydra. *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology.* **124**, S93 (1999).

9. Galliot, Brigitte *et al.*, Origins of neurogenesis, a cnidarian view. *Developmental Biology.* **332**, 2–24 (2009).

10. Miljkovic-Licina, Marijana, Gauchat, Dominique, Galliot, Brigitte, Neuronal evolution: analysis of regulatory genes in a first-evolved nervous system, the hydra nervous system. *Biosystems.* **76**, 75–87 (2004).

11. R. Augustin *et al.*, A secreted antibacterial neuropeptide shapes the microbiome of Hydra. *Nature*

communications. **8**, 698 (2017).

12. J. A. Chapman *et al.*, The dynamic genome of Hydra. *Nature*. **464**, 592–596 (2010).

13. H. Alexopoulos *et al.*, Evolution of gap junctions: the missing link? *Current biology : CB*. **14**, R879–80 (2004).