Description of the scHSQ package

UnJin Lee

January 7, 2025

Contents

1 Introduction

scHSQ is a package designed to implement identification of a gene set that passes the HSQ criteria as described in Lee, et al (2025).

A common issue of single cell transcriptomic analysis is in cell type identification across evolutionarily divergent species. Often, the transcriptomic divergence causes a strong species-specific effect that masks cell type effects such that cell types cannot be assigned across species. This is often true even when batch correction for species effect is applied.

Lee, et al (2025) describes a simple methodology for identifying genes that possess high cell type information while simultaneously being evolutionarily conserved. UMAP projections on these genes that pass the HSQ criteria, as opposed to the entire transcriptome, allow for accurate cell type assignments across multiple species. This is accomplished by performing an analysis of variance (ANOVA) for known cell type and species labels on a gene-wise basis.

While the identification of genes passing the HSQ criteria is relatively simple, this package simplifies process by utilizing *monocle3*. Additionally, it allows for the visualization of gene expression on a single cross-species UMAP projection, a task that may be difficult without use of this *scHSQ* package. All that is needed run this analysis is a *monocle3* data set, known cell type labels for a reference species, and labels for species-of-origin for all other cells.

2 Getting started

To install the package:

```
R CMD INSTALL scHSQ_x.y.z.tar.gz
```

scHSQ imports several functions from other packages. Make sure to have the following installed:
Biobase, *monocle3* and *ggplot2*.

3 First Steps

For demonstration purposes we use a test data set provided by this package. To do this, we must first import the library, then import the data set. It is important that the transcriptomes for each species be aligned to each other. In Lee, et al (2025), and thus the data set provided below, this is performed by retaining only one-to-one reciprocal best orthologs. This data set was produced using orthology information provided by FlyBase. Alternatively, other orthology-calling software may be used to perform this step, including the quick and accurate DIAMOND algorithm, e.g. Buchfink, Reuter, and Drost (2021).

```
> library(scHSQ)
> data(testis_3sp)
```

This data contains single-cell RNA-Seq data from three species: *D. melanogaster*, *D. yakuba*, and *D. ananassae*. The data is wrapped in a *monocle3* package with species information encoded in the `spec` field and cell type information encoded in the `known_type` field, with reference species labeled as `mel`. Each species' data set was individually imported using *monocle3*'s `new_cell_data_set` function, reduced to retain only one-to-one orthologs, then combined using *monocle3*'s `combine_cds` function. Subsequently, *monocle3*'s `preprocess_cds`, `reduce_dimension`, and `cluster_cells` functions were applied.

The resulting object is called `testis_3sp`. We thus use the known labels to generate a new *scHSQ* object as following:

```
> dros_testis <- new_scHSQ(testis_3sp, "mel", "spec", "known_type")
```

4 Applying HSQ Criteria

We now utilize the new *scHSQ* object to generate and apply our marker gene set. In this step, we calculate the ANOVA F-statistic for species and cell type on a gene-wise basis. We utilize median thresholds by default, but this may be altered either when initializing the new scHSQ object or through the use of `setPThresh`. This step may take a while, depending on the size of the input data sets.

```
> dros_testis <- genMarkGenes(dros_testis)
> dros_testis <- applyMarkGenes(dros_testis)
```

The cell type-specific divergence of each species' transcriptomes may be analyzed by plotting the *log(F-statistic)* values for each gene. A high correlation suggests a strong degree of cell type-specific evolution.

```
> plotANOVA(dros_testis)
```

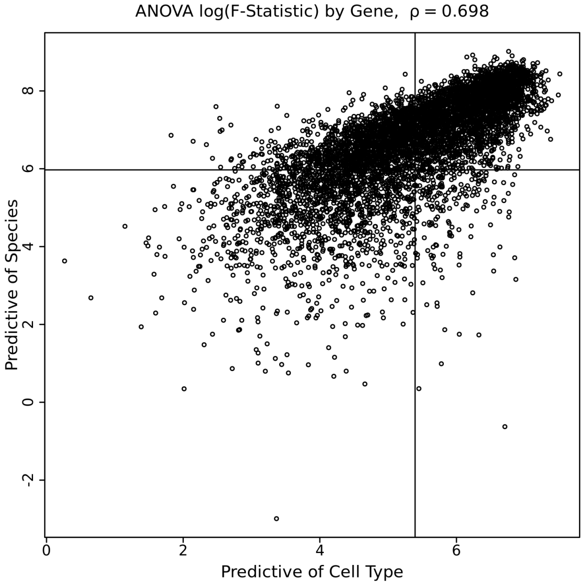

Figure 1: Strong cell type-specific evolution in *Drosophila* testis tissue.

After these commands complete, a cross-species UMAP projection should be stored in the `reducedCDS` slot of the *scHSQ* object. We can extract the list of marker genes utilized for this UMAP projection. Additionally, we can produce plots on the cross-species UMAP projection. In this case, the `ctplots` object is a list of UMAP projections containing: 1) known cell type labels for the reference species, 2) new clusters for all cells across all species, and 3) clusters for cells in each respective species. Note that arguments for *monocle3*'s `plot_cells` can be passed to *scHSQ*'s `plotCTAssign` function and will automatically be applied to all elements of `ctplots`.

```
> new_CDS <- getReducedCDS(dros_testis)
> gene_list <- getMarkGenes(dros_testis)
> ctplots <- plotCTAssign(dros_testis)
```

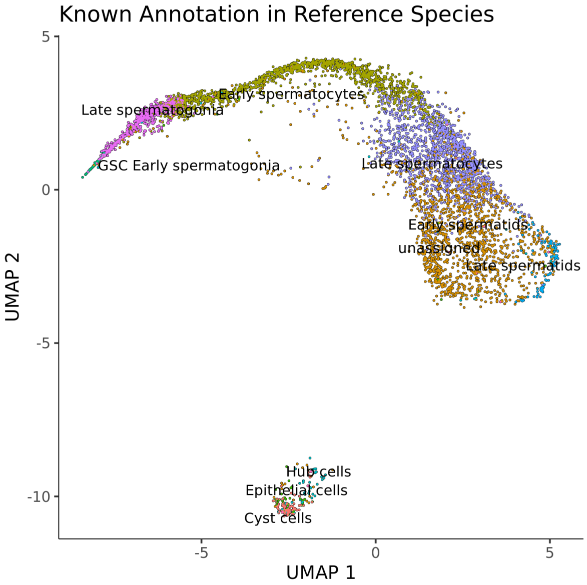

Figure 2: Single UMAP of original cell type assignments for reference species. This is found in `ctplots[[1]]`.

5 Cell type assignments across species

The UMAP figures from the previous section suggest a clear mapping of reference cell type assignments to the newly computed cell type assignments. If these mappings prove to be unsatisfactory, you may wish to adjust the HSQ thresholds using the `setPThresh` and/or recluster on the new UMAP with a different resolution using the `reclusterCDS` function. Once a satisfactory assignment has been achieved, the cell type assignments should be remapped using `remapCTAssign`. We may then plot the resulting re-assigned clusters.

```
> old_clu <- as.character(1:10)
> new_clu <- c("Late spermatocytes", "Early spermatocytes", "Early spermatids",
+ "Early spermatocytes", "Early spermatids", "Late spermatogonia", "Somatic",
+ "Late spermatids", "ananassae spermatocytes", "GSC Early spermatogonia")
> cell_types <- data.frame(HSQ_clu=old_clu, HSQ_clu1=new_clu)
> dros_testis <- remapCTAssign(dros_testis, cell_types)
> ctplots_reassign <- plotCTAssign(dros_testis, "HSQ_clu1")
```

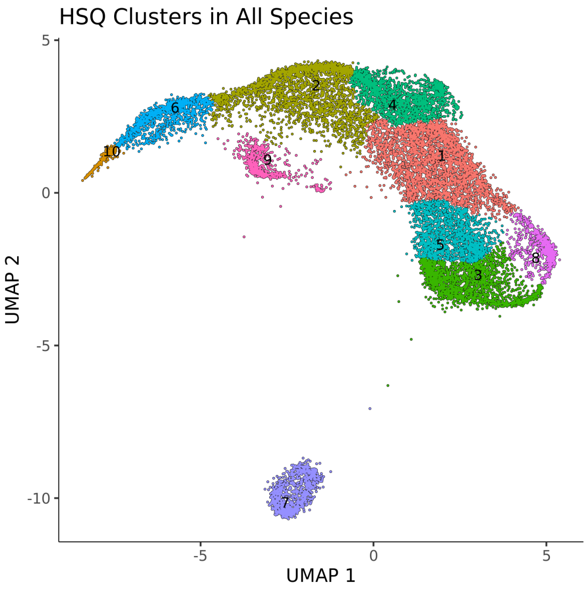

Figure 3: Single UMAP of new cell type assignments for all species. This is found in `ctplots[[2]]`.

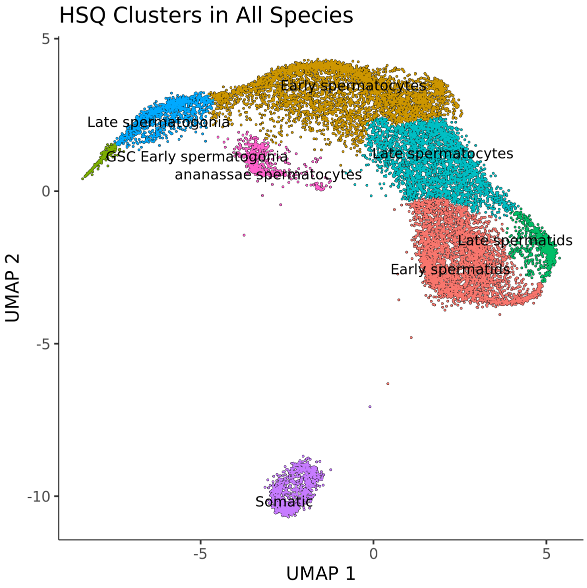

Figure 4: Single UMAP of re-assigned cell types for reference species. This is found in `ctplots_reassign[[2]]`.

6 Gene expression on new UMAP

Finally, we may be interested in seeing a single gene's expression on the new UMAP. To generate this figure, we use the example of *Rbp4*, a gene that did not pass the HSQ criteria. We utilize the `plotGeneExprs` function to generate this figure. As before, we are also able to forward arguments for *monocle3*'s `plot_cells` function. We may also generate this plot for a single species' cells by using the `species=` argument.

```
> gene_exprs <- plotGeneExprs(dros_testis, "Rbp4", alpha=0.5,
+ norm_method="size_only", min_expr=40)
```

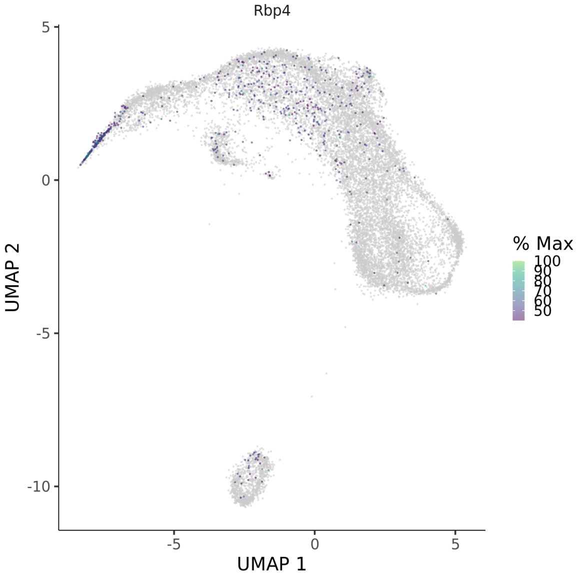

Figure 5: Expression of *Rbp4* on single UMAP of across species.

REFERENCES

Lee U, Li C, Langer CB, Svetec N, Zhao L (2025) Comparative Single Cell Analysis of Transcriptional Bursting Reveals the Role of Genome Organization on de novo Transcript Origination. bioRxiv doi:10.1101/2024.04.29.591771

Buchfink B, Reuter K, Drost H-G (2021), Sensitive protein alignments at tree-of-life scale using DIAMOND. Nature Methods, 18: 366-368 doi:10.1038/s41592-021-01101-x